Software Impacts 19 (2024) 100606

Contents lists available at ScienceDirect

Software Impacts

journal homepage: www.journals.elsevier.com/software-impacts

Original software publication

CAPRI: Context-aware point-of-interest recommendation framework

Ali Tourani [a], Hossein A. Rahmani [b,*], Mohammadmehdi Naghiaei [c], Yashar Deldjoo [d]

[a] University of Luxembourg, Luxembourg
[b] University College London, UK
[c] University of Southern California, USA
[d] Polytechnic University of Bari, Italy

ARTICLE INFO

Keywords:
Recommender systems
Software tools
Framework
Reproducibility

ABSTRACT

Point-of-Interest (POI) recommendation systems have gained popularity for their unique ability to suggest geographical destinations, with the incorporation of contextual information such as time, location, and user-item interaction. Existing recommendation frameworks lack the contextual fusion required for POI systems. This paper presents CAPRI, a novel POI recommendation framework that effectively integrates context-aware models, such as GeoSoCa, LORE, and USG, and introduces a novel strategy for the efficient merging of contextual information. CAPRI integrates an evaluation module that expands the evaluation scope beyond accuracy to include novelty, personalization, diversity, and fairness. With an aim to establish a new industry standard for reproducible results in the realm of POI recommendation systems, we have made CAPRI openly accessible on GitHub, facilitating easy access and contribution to the continued development and refinement of this innovative framework.

Code metadata

Current code version	*v0.1*
Permanent link to code/repository used for this code version	https://github.com/SoftwareImpacts/SIMPAC-2023-444
Permanent link to Reproducible Capsule	https://codeocean.com/capsule/6574520/tree/v1
Legal Code License	*MIT License*
Code versioning system used	*Git, GitHub*
Software code languages, tools, and services used	*Python, Jupyter Notebook*
Compilation requirements, operating environments & dependencies	
If available Link to developer documentation/manual	https://caprirecsys.github.io/CAPRI/
Support email for questions	caprirecsys@gmail.com

1. Introduction and context

For selecting the appropriate vacation destination, restaurant, visiting locations, and so-called Point-of-Interest (POI), users must choose from a variety of possibilities. POI recommender systems can be useful tools for overcoming the inevitable information overload in many use cases. For instance, recommending hotels and other travel-related destinations remains a challenging task since trip planning entails looking for a set or list of interconnected factors (*e.g.,* means of transportation, housing, and attractions) and where contextual factors may have a significant impact (*e.g.,* time, location, and social environment).

Recent years have seen the development of numerous frameworks, libraries, and tools for Recommender System (RS) that make it easy for researchers to mimic the recommendation process and its influence on user preference. Utilizing frameworks for recommendation leads to the standardization of algorithm implementations and facilitates the reproducibility of experiments. Despite the advances, reproducibility remains a challenge in RS research, particularly in the areas that are not well-established, such as fairness-aware and domain-specific recommendations [1]. Even minor differences in parameters and experimental settings can yield inconsistent results, making it difficult

The code (and data) in this article has been certified as Reproducible by Code Ocean: (https://codeocean.com/). More information on the Reproducibility Badge Initiative is available at https://www.elsevier.com/physical-sciences-and-engineering/computer-science/journals.

* Corresponding author.
E-mail addresses: ali.tourani@uni.lu (A. Tourani), hossein.rahmani.22@ucl.ac.uk (H.A. Rahmani), naghiaei@usc.edu (M. Naghiaei), yashar.deldjoo@poliba.it (Y. Deldjoo).

https://doi.org/10.1016/j.simpa.2023.100606
Received 14 November 2023; Accepted 28 November 2023

to provide definitive answers about the relative properties of different algorithms. Hence, reproducible evaluation and fair comparison of methods are demanding factors in RSs.

Despite the progress made in the field, existing frameworks for the reproducibility of RSs are typically intended to simulate generic Collaborative Filtering (CF) environments. For instance, `Cornac` [2] includes models leveraging auxiliary data such as item descriptive text and image. `RecBole` [3], an alternative comprehensive framework, introduces general, sequential, and knowledge-based recommendations. Likewise, `Elliot` [4] is another framework that covers a wide range of general-purpose models.

However, POI recommendation has particularities that set them apart from recommendations in other domains:

- **The importance of context integration and fusion**: The users' check-ins in POI recommendations are considerably affected by the contextual information. For instance, the geographical property of location affects the user mobility pattern or users' visit is time-depended which indicates the importance of temporal information [5]. Other types of context may include social ties, the category of POIs, comments on POIs, *etc.* Previous research works such as [6] have shown the way incorporating these **rich contexts** information have a significant impact on the performance of POI recommendation models. Therefore, in recent years, there has been a growth in the demand for specialized recommendation algorithms and methodologies that can **incorporate** and **fuse contextual** information into the POI recommendation process [7];
- **High sparsity**: The characteristics of the check-in datasets of the POI recommendation domain differ significantly from those of the other recommendation domain [8,9]. Accordingly, the density of POI check-ins data is typically approximately 0.1%, whereas the density of *Netflix* data for movie suggestions is 1.2%. This is because a person can only visit a limited number of locations, whereas a city can contain a vast number of POIs;
- **Necessity for multi-dimensional evaluation**: Previous papers [9–11] in the POI field predominantly focus on accuracy-oriented metrics. However, there is a remarkable consensus in the RS community that there are other important facets to the recommendation process that accuracy metric systems cannot simply capture, such as the novelty, diversity, and catalog coverage of recommenders. Therefore, we aim to standardize multi-faceted evaluation on the accuracy, beyond-accuracy, and fairness dimensions [12].

Contributions. The work at hand addresses the above shortcoming by proposing *`CAPRI`*,[1] a specialized framework for evaluating and benchmarking state-of-the-art POI recommendation models. Different from existing open-source frameworks, such as `DaisyRec` [13], `Elliot` [4], `LensKit` [14], `LibRec` [15], `LibRec-auto` [16], `OpenRec` [17], `CaseRec` [18], which mainly aim to reproduce various traditional recommender systems, deep learning-based recommender systems such as `DeepRec` [19], and multimodal RSs like `Cornac` [2], `CAPRI` is intended to provide contextually aware recommendation and evaluation in the POI domain. We have equipped our framework with state-of-the-art models, algorithms, well-known datasets for POI recommendations, and multi-dimensional evaluation criteria (accuracy, beyond-accuracy, and user-item fairness). It also supports the reproducibility of results using various adjustable configurations for comparative experiments.

To the best of our knowledge, there is no publicly accessible framework for the reproducibility of POI models in the field of context-aware POI recommendation, despite the recent advances in the field.

The majority of current frameworks are general-purpose and do not prioritize domain-specific recommendation models, such as context awareness. This characteristic makes it challenging to re-purpose their research skills for domain-specific work. In contrast to the introduced frameworks, `CAPRI` focuses on the POI domain and aims to provide researchers with all the necessary resources.

2. Proposed framework

`CAPRI` is an open-source recommendation framework implemented in Python, suitable for practical experimentation and reproducibility study. The framework is distributed under the *GPL v.3* license and can be downloaded or cloned from GitHub.

2.1. Files structure

In terms of implementation, the files of the framework are organized in several directories to facilitate accessibility and extensibility. We utilize *PascalCase* and *camelCase* as basic naming structures and merge words into a single string in `CAPRI` for folder and file names, respectively. Detailed descriptions of the directories of the framework containing files are presented below in brief:

- **Data:** Contains data-driven files and functions of various types. Each dataset includes files with the `.txt` extension that contain train, test, and tune data. Moreover, other files containing the check-ins data and relations among users/items, such as social and geographical data, are stored in folders with the same name as each dataset. There are also some data processing functions in the Data directory, including `readDataSizes.py` to read meta-data of the dataset, `loadDatasetFiles.py` to load selected dataset items, and `calculateActiveUsers.py` to calculate Active/Inactive users of a selected dataset for fairness-aware analysis. Current datasets of `CAPRI` will be discussed in Section 2.2.
- **Models:** Contains the models used in the framework and several common functions in the `utils.py` file to avoid code duplication and increase the re-usability of model files. For each model, there is a folder with the same name, a `main.py` to control the overall processing of functions, and varying processing functions to process a selected dataset according to the selected model. The accessible models in `CAPRI` will be discussed in more depth in Section 2.3.
- **Evaluations:** Contains all evaluation metrics available for analyzing the performance of models on datasets, the evaluator function `evaluator.py` that leverages the metrics, and a unit test file `test.py` for evaluating the performance of each measure with different input types. The evaluation metrics supplied in `CAPRI`, as well as the evaluation process, will be discussed in detail in Sections 2.4 and 3, respectively.
- **Outputs:** `CAPRI` stores the final findings, including ranked lists and evaluation outputs, for reproducibility purposes. The file naming structure prohibits a previously performed analysis from being reprocessed. It should be noted that due to the size of the ranked list files, we do not save them on GitHub.

Regarding the fact that `CAPRI` is open source and accepts contributions, it is straightforward to add new models, datasets, and metrics. We welcome academics and developers who aim to contribute to the framework's enhancement. Consequently, documentation on how to contribute to the project is available at the *readthedocs* page of the framework.[2]

[1] https://caprirecsys.github.io/CAPRI/

[2] https://capri.readthedocs.io/en/latest/

Table 1
Characteristics of the datasets available in CAPRI.

Dataset	#users	#POIs	#check-ins	#social	#category
Yelp	7,135	15,575	299,327	46,778	582
Gowalla	5,628	30,943	618,621	46,001	–
Foursquare	7,642	28,483	512,532	–	–

2.2. Datasets

In the current version of the framework, we have provided modified versions of three popular check-ins datasets: Gowalla,[3] Yelp[4] [9], and Foursquare.[5] The characteristics of the mentioned datasets are presented in Table 1.

2.3. Models

CAPRI covers the recent implementations of various models, which can be applied to the introduced datasets for evaluation and reproducibility goals. The models implemented in this framework are listed below:

- **GeoSoCa:** As introduced in [20], this model covers geographical, social, and categorical correlations among users and POIs. These contexts are learned using users' historical check-in data to produce relevance scores for unseen locations.
- **LORE:** Another model utilized in CAPRI is LORE [21], a popular and robust model for location recommendation focused on the impacts of geographical and social influence on users' check-in behaviors.
- **USG:** As introduced in [22], USG takes geographical influence, social network, and user interest into account for POI recommendation.

The current CAPRI version covers standard competitive contextual models for the POI domain, with users having the flexibility to modify contexts per their requirements. Our future plans include the incorporation of deep learning, graph-based, sequential, and sessions-based models as proposed in works like [23,24]. These models can integrate various contextual components like geographical, temporal, social, and categorical relevance scores using fusion rules such as *product* or *sum* [6,9], forming a unified preference score [25–27]. Contextual information, denoted by c_i, can be infused using a polynomial regression model

$$rec_{u,p} = \mathbf{\Lambda} \cdot \mathbf{C} + \mathbf{\Lambda}_{\text{pair}} \cdot \mathbf{C}_{\text{pair}} + \lambda_{123} c_1 c_2 c_3 \tag{1}$$

where:

- $\mathbf{\Lambda} = [\lambda_1, \lambda_2, \lambda_3]$
- $\mathbf{C} = [c_1, c_2, c_3]$
- $\mathbf{\Lambda}_{\text{pair}} = [\lambda_{12}, \lambda_{13}, \lambda_{23}]$
- $\mathbf{C}_{\text{pair}} = [c_1 c_2, c_1 c_3, c_2 c_3]$

in which λ_j indicates the importance weight for the context c_i learned by the model. Note that the product rule ($\bigodot$) would have $\lambda_j = 0$ for all j and $\lambda_{123} = 1$. In the case of the sum ($\bigoplus$), λ_1, λ_2, and λ_3 are 1 and the rest are 0, while in the weighted sum ($\boxplus$), optimal values are assigned to them to maximize performance criteria.

2.4. Evaluation dimensions

CAPRI is highly compatible with a range of evaluation metrics. Accordingly, the evaluation metrics available in the framework can be classified into the following categories:

- **Accuracy:** for accuracy evaluation, the framework covers *Precision@k*, *Recall@k*, *mAP@k*, and *nDCG@k* metrics, in which k represents the number of items filtered for recommendation.
- **Beyond-Accuracy:** this category contains *List Diversity*, *Novelty*, *Catalog Coverage*, and *Personalization* metrics.
- **Fairness:** It contains modules for grouping users and items according to a sensitive attribute. Thus, it includes *MADr* and *GCE* evaluation among the user/item groups [12,28,29].

It is observable that the offered metrics are tailored to meet the recommendations of POI recommendation. All the evaluation metrics can be accessed using the `Evaluations` directory in the framework. There is also a `test.py` file in the same folder for evaluating the performance of each metric through unit testing.

2.5. Configuration

CAPRI contains a `config.py` file that provides adjustments and configurations for running various experiments. Accordingly, the parameters that can be set using the mentioned file are listed below:

- `dataDirectory`: the path from which the dataset files are read,
- `outputsDir`: the path to store final recommendation lists generated by the framework,
- `topK`: *Top-k* items for doing the evaluations (default: 10),
- `limitUsers`: limiting the number of users in the dataset (default: -1),
- `listLimit`: limiting the length of the final recommendation lists (default: 10),
- `activeUsersPercentage`: calculating a list of pre-defined groups of users known as *"active users"*,
- `models`: available models to be selected by the user,
- `datasets`: available datasets to be selected by the user,
- `fusions`: available fusions to be selected by the user,
- `evaluationMetrics`: available evaluation metrics to be selected by the user.

3. Evaluation process

This section describes how the evaluation process takes place in CAPRI. All the evaluation-related functions of the framework are collected in `evaluator.py` file in the *Evaluations* directory. Accordingly, the model and the dataset selected by the user, as well as the evaluation and model parameters, are passed to the evaluator. Model parameters contain model-specific final scores, such as geographical, social, and categorical calculations for GeoSoCa. Evaluation parameters are formed as a dictionary that contains evaluation-related feed, such as the list of users, the list of POI, and ground truth data. By iterating over users in the dataset, overall recommendation scores and requested accuracy measures such as *Precision@k* and *Recall@k* are calculated. The final results will be saved in files for later processes.

4. Benchmarking

Table 2 shows initial and experimental results using our proposed framework, CAPRI. As one can see, using CAPRI, we are able to incorporate different contextual models of the POI recommendation domain as well as different approaches to context fusion. We can see that the fusion methods have a great impact on the performance of POI models. The *sum* rule could show a much better impact on the beyond-accuracy performances (*i.e.,* coverage, novelty, and diversity) compared to the *product* rule. Additionally, it can be seen that the *sum* rule often outperforms the *product* operation in accuracy for the GeoSoCa model. The *product* operation often outperforms the *sum* operation in accuracy for LORE (with the exception of *Recall* where the *sum* is better than the *product*). Finally, the *weighted sum* has a favorable effect on the

[3] http://www.gowalla.com/
[4] https://www.yelp.com/dataset
[5] https://sites.google.com/site/yangdingqi/home/foursquare-dataset

Table 2
Accuracy and beyond-accuracy performance of models and baselines evaluated on top-20 recommendation lists on the Yelp dataset. In context, g, t, s, and c show the geographical, temporal, social, and categorical contexts, respectively. (Bold the best metric values).

Model	Context				Fusion	Accuracy			Beyond-accuracy			Fairness
	g	t	s	c		*nDCG*	*Precision*	*Recall*	*Coverage*	*Novelty*	*Diversity*	*MADr*
MostPop	–	–	–	–	–	0.0073	0.0064	0.0185	0.12	4.9590	0.6938	**0.0152**
MF	–	–	–	–	–	0.0078	0.0070	0.0181	0.3	4.9736	0.6877	0.0219
GeoSoCa	✓	×	✓	✓	⊙	0.0169	0.0134	0.0341	49.4	7.9083	0.6877	0.0443
					⊕	0.0160	0.0138	0.040	73.26	**8.5047**	0.7166	0.0525
					⊞	0.0170	**0.0145**	**0.0419**	69.75	8.3311	0.6835	0.0465
LORE	✓	✓	✓	×	⊙	**0.0174**	0.0139	0.0335	39.14	7.7314	0.7509	0.0530
					⊕	0.0155	0.0134	0.0388	**74.97**	8.4723	**0.8065**	0.0404
					⊞	0.0169	0.0143	0.0412	74.51	8.1677	0.7971	0.0404

GeoSoCa model, making it the most accurate model and enhancing the *sum* model under LORE. As a work where CAPRI has been employed for POI recommendation, further evaluation of the results for user/item fairness can be found in [30].

5. Impact overview

By employing the context-aware recommendation offered by the CAPRI [11], recommendations can be evaluated and become context-aware to improve the performance and the quality of recommendation and personalization. As an example, Rahmani et al. [31] used CAPRI to evaluate the role of context fusion on accuracy, beyond-accuracy, and fairness (from user and item perspective) of point-of-interest recommendation systems. Rahmani et al. [27] used CAPRI to provide an analysis and exploration of the impact of temporal bias in Point-of-Interest recommendation. As a future development, it is possible to incorporate other notions of contextual fairness (such as calibration based on temporal aspects) and recommendation novelty/diversity. These are additional research objectives that may be considered in conjunction with more contextual information and fusion ways.

6. Discussion and conclusion

The open-source framework, CAPRI, developed for POI recommendation systems, distinguishes itself by leveraging contextual information in the suggestion pipeline. It incorporates robust models, datasets, and evaluation metrics to provide proper location suggestions matching the context. Like any software, constant development is necessary for CAPRI to accommodate user needs. We plan on widening its coverage of datasets and models, incorporating more evaluation metrics, and investigating parallelization and requests queuing to improve performance. Future iterations will support batch request handling with user-defined parameters and offer a GUI for easier configuration. We aim to make it directly installable via Python's pip, and integrate bias mitigation approaches to reduce system biases. The framework is publicly available on GitHub for researchers.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

References

[1] Nasim Sonboli, Robin Burke, Zijun Liu, Masoud Mansoury, Fairness-aware recommendation with librec-auto, in: Proceedings of the 14th ACM Conference on Recommender Systems, RecSys '20, Association for Computing Machinery, New York, NY, USA, 2020, pp. 594–596.

[2] Aghiles Salah, Quoc-Tuan Truong, Hady W. Lauw, Cornac: A comparative framework for multimodal recommender systems, J. Mach. Learn. Res. 21 (2020) 91–95.

[3] Wayne Xin Zhao, Shanlei Mu, Yupeng Hou, Zihan Lin, Yushuo Chen, Xingyu Pan, Kaiyuan Li, Yujie Lu, Hui Wang, Changxin Tian, et al., Recbole: Towards a unified, comprehensive and efficient framework for recommendation algorithms, in: Proceedings of the 30th ACM International Conference on Information & Knowledge Management, Association for Computing Machinery, New York, NY, USA, 2021, pp. 4653–4664.

[4] Vito Walter Anelli, Alejandro Bellogin, Antonio Ferrara, Daniele Malitesta, Felice Antonio Merra, Claudio Pomo, Francesco Maria Donini, Tommaso Di Noia, Elliot: A comprehensive and rigorous framework for reproducible recommender systems evaluation, in: Proceedings of the 44th International ACM SIGIR Conference on Research and Development in Information Retrieval, SIGIR '21, Association for Computing Machinery, New York, NY, USA, 2021, pp. 2405–2414.

[5] Hossein A. Rahmani, Mohammad Aliannejadi, Mitra Baratchi, Fabio Crestani, Joint geographical and temporal modeling based on matrix factorization for point-of-interest recommendation, in: European Conference on Information Retrieval, Springer, Cham, 2020, pp. 205–219.

[6] Hossein A. Rahmani, Mohammad Aliannejadi, Mitra Baratchi, Fabio Crestani, A systematic analysis on the impact of contextual information on point-of-interest recommendation, ACM Trans. Inf. Syst. 40 (4) (2022).

[7] Heitor Werneck, Nícollas Silva, Adriano Pereira, Matheus Carvalho, Alejandro Bellogín, Jorge Martinez-Gil, Fernando Mourão, Leonardo Rocha, A reproducible POI recommendation framework: Works mapping and benchmark evaluation, Inf. Syst. 108 (2022) 102019.

[8] Robert M. Bell, Yehuda Koren, Lessons from the netflix prize challenge, Acm Sigkdd Explor. Newsl. 9 (2) (2007) 75–79.

[9] Yiding Liu, Tuan-Anh Nguyen Pham, Gao Cong, Quan Yuan, An experimental evaluation of point-of-interest recommendation in location-based social networks, Proc. VLDB Endow. 10 (10) (2017) 1010–1021.

[10] Heitor Werneck, Nícollas Silva, Matheus Carvalho, Adriano CM Pereira, Fernando Mourão, Leonardo Rocha, A systematic mapping on POI recommendation: Directions, contributions and limitations of recent studies, Inf. Syst. 101 (2021) 101789.

[11] Pablo Sánchez, Alejandro Bellogín, Point-of-interest recommender systems based on location-based social networks: A survey from an experimental perspective, ACM Comput. Surv. 54 (11s) (2022).

[12] Yifan Wang, Weizhi Ma, Min Zhang, Yiqun Liu, Shaoping Ma, A survey on the fairness of recommender systems, ACM Trans. Inf. Syst. 41 (3) (2023).

[13] Zhu Sun, Di Yu, Hui Fang, Jie Yang, Xinghua Qu, Jie Zhang, Cong Geng, Are we evaluating rigorously? Benchmarking recommendation for reproducible evaluation and fair comparison, in: Proceedings of the 14th ACM Conference on Recommender Systems, RecSys '20, Association for Computing Machinery, New York, NY, USA, 2020, pp. 23–32.

[14] Michael D. Ekstrand, Michael Ludwig, Joseph A. Konstan, John T. Riedl, Rethinking the recommender research ecosystem: Reproducibility, openness, and LensKit, in: Proceedings of the Fifth ACM Conference on Recommender Systems, RecSys '11, Association for Computing Machinery, New York, NY, USA, 2011, pp. 133–140.

[15] Guibing Guo, Jie Zhang, Zhu Sun, Neil Yorke-Smith, Librec: A java library for recommender systems, in: UMAP Workshops, Vol. 4, Citeseer, Citeseer, 2015, pp. 38–45.

[16] Nasim Sonboli, Masoud Mansoury, Ziyue Guo, Shreyas Kadekodi, Weiwen Liu, Zijun Liu, Andrew Schwartz, Robin Burke, Librec-auto: A tool for recommender systems experimentation, in: Proceedings of the 30th ACM International Conference on Information &Amp; Knowledge Management, Association for Computing Machinery, New York, NY, USA, 2021, pp. 4584–4593.

[17] Longqi Yang, Eugene Bagdasaryan, Joshua Gruenstein, Cheng-Kang Hsieh, Deborah Estrin, OpenRec: A modular framework for extensible and adaptable recommendation algorithms, in: Proceedings of the Eleventh ACM International Conference on Web Search and Data Mining, WSDM '18, Association for Computing Machinery, New York, NY, USA, 2018, pp. 664–672.

[18] Arthur da Costa, Eduardo Fressato, Fernando Neto, Marcelo Manzato, Ricardo Campello, Case recommender: A flexible and extensible python framework for recommender systems, in: Proceedings of the 12th ACM Conference on Recommender Systems, RecSys '18, Association for Computing Machinery, New York, NY, USA, 2018, pp. 494–495.

[19] Wen Zhang, Yuhang Du, Taketoshi Yoshida, Ye Yang, DeepRec: A deep neural network approach to recommendation with item embedding and weighted loss function, Inf. Sci. 470 (2019) 121–140.

[20] Jia-Dong Zhang, Chi-Yin Chow, GeoSoCa: Exploiting geographical, social and categorical correlations for point-of-interest recommendations, in: Proceedings of the 38th International ACM SIGIR Conference on Research and Development in Information Retrieval, SIGIR '15, Association for Computing Machinery, New York, NY, USA, 2015, pp. 443–452.

[21] Jia-Dong Zhang, Chi-Yin Chow, Yanhua Li, LORE: Exploiting sequential influence for location recommendations, in: Proceedings of the 22nd ACM SIGSPATIAL International Conference on Advances in Geographic Information Systems, SIGSPATIAL '14, Association for Computing Machinery, New York, NY, USA, 2014, pp. 103–112.

[22] Mao Ye, Peifeng Yin, Wang-Chien Lee, 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, SIGIR '11, Association for Computing Machinery, New York, NY, USA, 2011, pp. 325–334.

[23] Jens Adamczak, Yashar Deldjoo, Farshad Bakhshandegan Moghaddam, Peter Knees, Gerard-Paul Leyson, Philipp Monreal, Session-based hotel recommendations dataset: As part of the ACM recommender system challenge 2019, ACM Trans. Intell. Syst. Technol. 12 (1) (2020).

[24] Massimo Quadrana, Alexandros Karatzoglou, Balázs Hidasi, Paolo Cremonesi, Personalizing session-based recommendations with hierarchical recurrent neural networks, in: Proceedings of the Eleventh ACM Conference on Recommender Systems, RecSys '17, Association for Computing Machinery, New York, NY, USA, 2017, pp. 130–137.

[25] Hossein A. Rahmani, Mohammad Aliannejadi, Sajad Ahmadian, Mitra Baratchi, Mohsen Afsharchi, Fabio Crestani, LGLMF: local geographical based logistic matrix factorization model for POI recommendation, in: Asia Information Retrieval Symposium, Springer, Cham, 2019, pp. 66–78.

[26] Hossein A. Rahmani, Mohammad Aliannejadi, Rasoul Mirzaei Zadeh, Mitra Baratchi, Mohsen Afsharchi, Fabio Crestani, Category-aware location embedding for point-of-interest recommendation, in: Proceedings of the 2019 ACM SIGIR International Conference on Theory of Information Retrieval, ICTIR '19, Association for Computing Machinery, New York, NY, USA, 2019, pp. 173–176.

[27] Hossein A. Rahmani, Mohammadmehdi Naghiaei, Ali Tourani, Yashar Deldjoo, Exploring the impact of temporal bias in point-of-interest recommendation, in: Proceedings of the 16th ACM Conference on Recommender Systems, RecSys '22, Association for Computing Machinery, New York, NY, USA, 2022, pp. 598–603.

[28] Yashar Deldjoo, Vito Walter Anelli, Hamed Zamani, Alejandro Bellogin, Tommaso Di Noia, A flexible framework for evaluating user and item fairness in recommender systems, User Model. User-Adapt. Interact. 31 (3) (2021) 457–511.

[29] Hossein A. Rahmani, Yashar Deldjoo, Ali Tourani, Mohammadmehdi Naghiaei, The unfairness of active users and popularity bias in point-of-interest recommendation, in: Advances in Bias and Fairness in Information Retrieval: Third International Workshop, BIAS 2022, Stavanger, Norway, April 10, 2022, Revised Selected Papers, Springer, Switzerland, 2022, pp. 56–68.

[30] Hossein A. Rahmani, Yashar Deldjoo, Tommaso Di Noia, The role of context fusion on accuracy, beyond-accuracy, and fairness of point-of-interest recommendation systems, Expert Syst. Appl. 31 (3) (2022) 457–511.

[31] Hossein A. Rahmani, Yashar Deldjoo, Tommaso Di Noia, The role of context fusion on accuracy, beyond-accuracy, and fairness of point-of-interest recommendation systems, Expert Syst. Appl. 205 (2022) 117700.