Smart imputation, better recommendations: improving traditional Point-of-Interest recommendation through data augmentation

PABLO SÁNCHEZ, Instituto de Investigación Tecnológica (IIT), Universidad Pontificia Comillas, Spain
ALEJANDRO BELLOGÍN, Universidad Autónoma de Madrid, Spain

Data sparsity is a persistent challenge in recommender systems, specially in specific domains like Point-of-Interest (POI) recommendation, where it significantly impacts model performance. While classical recommender systems have used various imputation and data augmentation mechanisms to address data sparsity, these methods have not been extensively explored in the POI recommendation domain. In this work, we propose a generic imputation framework to study the use of data augmentation techniques to generate synthetic check-ins and analyze their effects on the POI recommendation scenario. Our main goal is to enhance the performance of various traditional recommenders by increasing the training set interactions, considering specific characteristics of the domain, such as geographical information. We apply these techniques in six different cities from a global Foursquare check-in dataset, as well as in two additional cities from the Gowalla dataset, and a separate dataset from Yelp, ensuring a comprehensive evaluation across multiple data sources. Our imputation approach evidences improvements for most models. In several cases, these improvements exceeded 100% for ranking accuracy, measured in terms of nDCG, without considerably compromising novelty or diversity. Data and code is released at https://github.com/pablosanchezp/ImputationForPOIRecsys.

CCS Concepts: • **Information systems** → **Recommender systems**; Information extraction.

Additional Key Words and Phrases: Point-Of-Interest, Imputation, Temporal evaluation

ACM Reference Format:
Pablo Sánchez and Alejandro Bellogín. 2025. Smart imputation, better recommendations: improving traditional Point-of-Interest recommendation through data augmentation. 1, 1 (June 2025), 34 pages. https://doi.org/10.1145/nnnnnnn.nnnnnnn

1 INTRODUCTION

Recommender Systems (RSs) have become indispensable in various economic sectors, including online retail, entertainment, and even online dating [50]. These systems suggest different items, e.g., products, movies, TV series, or potential matches based on user preferences and behaviors. As technology advances, the role of RSs in shaping user experiences across different domains continues to expand, highlighting their importance in modern digital interactions. One of the most representative examples is the tourism industry, where RSs also play an important role, as demonstrated by the substantial research dedicated to the Point-of-Interest (POI) recommendation problem and other related tasks [53, 36]. The main goal in the POI recommendation problem is to suggest new venues (or POIs) to be visited by the users when they are in a city. The information used to train the models

Authors' Contact Information: Pablo Sánchez, psperez@icai.comillas.edu, Instituto de Investigación Tecnológica (IIT), Universidad Pontificia Comillas, Madrid, Spain; Alejandro Bellogín, alejandro.bellogin@uam.es, Universidad Autónoma de Madrid, Madrid, Spain.

ACM XXXX-XXXX/2025/6-ART
https://doi.org/10.1145/nnnnnnn.nnnnnnn

is usually obtained from Location-based Social Networks (LBSNs) such as Gowalla, Foursquare, or Yelp. In these social networks, users register the check-ins they perform in the different POIs they visit, and sometimes they establish friendship links with each other. Using this information, it is possible to infer users' preferences and interests in order to generate recommendations on new venues to visit [40]. By analyzing user data and preferences, POI recommenders enhance the travel experience, offering personalized suggestions for attractions, restaurants, activities, etc. This personalized approach not only improves user satisfaction but also drives engagement and revenue in the tourism sector [1, 23].

The main objective of Point-of-Interest recommendation is the same as that of traditional recommendations: to provide users with items that are particularly interesting to them. Nevertheless, this domain has specific characteristics that must be considered when modeling the recommendation task. For instance, temporal and sequential influences – crucial in other domains like music [43] – are relevant here, as we can track the routes users take when visiting POIs. However, the most critical factor is geographical influence, as users generally prefer venues near their current location [53]. Unlike the traditional recommendation scenario where users typically rate each item only once, in POI recommendation it is common for users to visit the same POIs multiple times. Hence, instead of using the traditional user $\times$ item rating matrix, most researchers make use of frequency matrices, where each value represents the number of visits a user has made to a particular venue. As a consequence, data sparsity is particularly severe in the POI recommendation domain.

While sparsity is a significant challenge in traditional recommender systems (for instance, the Movielens25M dataset has a sparsity of 99.75%), it is even more pronounced in POIs datasets. Two of the most widely used datasets for POI recommendation, Gowalla and Foursquare [53, 35, 11], exhibit sparsity levels of 99.9966% and 99.9953%, respectively. These low density levels are a challenge in the recommendation process and reduces the model performance [34, 57]. To alleviate these sparsity issues, most state-of-the-art approaches take advantage of various of the aforementioned influences, including geographical, social, temporal, and POIs categories, to generate better recommendations [44]. However, the performance of these models remains significantly lower than in traditional recommendation scenarios. In fact, one approach that has been studied in classical recommendation domains, but not in detail in POI recommendation, is the so-called data imputation or augmentation [3, 48, 25]. These methods allow recommendation algorithms to consider extended training data, where some (or all) missing values have been inferred, thereby reducing the input sparsity the algorithms need to face. As a result, the more (useful) information that the recommenders have, the better performance they normally achieve [12].

Data augmentation in POI recommendation presents a unique challenge, particularly when it comes to generating new interactions that can provide recommender systems with more useful information. The core of the issue lies in the uncertainty of how to effectively perform such data imputation. Specifically, it is unclear which users should be targeted for these imputed or synthetic interactions, or which mechanisms should be used for generating these interactions. Moreover, it is crucial to define the requirements that the imputed interactions need to meet (considering, for example, restrictions regarding the geographical location of the users). Thus, the selection process must be carefully designed to ensure that the augmented data is realistic and beneficial to the learning model of the process. For that reason, just as in traditional recommender systems where denser datasets generally yield better results [12], we believe that the performance of the models in POI recommendation will also improve if we impute useful data.

Our work. To address the problem of data sparsity in POI recommendation, in this paper we propose to integrate, adapt, and exploit data imputation techniques to increase the available check-ins for the recommenders, with the purpose of enhancing their overall performance in the POI recommendation domain. Our proposed framework defines two sets of parameters that allow us to

model the necessary components for the successful imputation of new and relevant data to the recommendation methods, which correspond to check-ins instead of ratings. These parameters include determining the percentage of interactions to be imputed, identifying the specific users and items for imputation, and considering any additional domain-dependent restrictions that may influence the imputation process (e.g., considering geographical constraints). Moreover, our proposal is aligned with the existing literature on this topic, and we argue that future algorithmic developments could greatly benefit from such an approach.

Research questions. In order to validate our proposal, we analyze the performance of various well-known traditional recommender systems focusing on a global-scale Foursquare check-in dataset, although we have also considered other LBSN datasets (i.e., Gowalla and Yelp) in our experiments. We perform a comprehensive offline temporal evaluation across multiple cities to evaluate the effectiveness of the recommenders and address the following research questions: (**RQ1**) What are the required building blocks for an imputation framework tailored for Point-of-Interest recommendation? In particular, which dimensions or parameters should be considered? (**RQ2**) By applying this imputation framework, is it possible to improve ranking accuracy? What about other evaluation dimensions like novelty and diversity? (**RQ3**) Which parameters of this framework are more critical to achieve performance improvements?

Specifically, the **main contributions** of our work can be summarized as follows:

- We propose a novel and generic framework that enables the definition of a set of parameters to guide the data imputation process for POI recommendation, thereby enhancing the volume of data available in the training set and reducing the data sparsity. By augmenting the available data, this framework aims to improve the performance of both traditional and POI recommender systems.
- We define a set of methodological settings designed to fairly and appropriately evaluate recommendation algorithms with and without imputed data. By exploiting such settings, we could derive correct and replicable conclusions from the obtained results.
- We present a thorough experimental comparison across six different cities from the Foursquare dataset belonging to various continents, two additional cities from the Gowalla dataset, and a separate dataset from Yelp. We use a temporal offline evaluation testing methodology to assess well-known recommendation algorithms from both traditional and POI recommendation domains. The results report improved values not only in terms of ranking accuracy but also in terms of novelty and diversity.

Implications. As our results shall demonstrate, the highly configurable imputation framework we propose allows the incorporation of various parameters to enhance data quality, leading to competitive ranking accuracy across datasets from different sources and with varying inherent characteristics. This supports the premise that the performance of POI recommenders can be improved by reducing the training sparsity through the imputation of meaningful check-ins. Additionally, our proposal is sufficiently generic, allowing other researchers to integrate it into their models and achieve an improved performance in other domains.

The remainder of the paper is organized as follows: in Section 2, we introduce the problem of POI recommendation and summarize related work on different techniques that have been proposed to reduce the data sparsity existing in recommender systems. In Section 3 we present in detail the imputation framework we propose and provide some informative examples. Section 4 describes the evaluation methodology followed in the experiments presented in Section 5, where we show the results obtained when comparing the performance of the recommenders without any type of data imputation versus those obtained using various instantiations of our imputation framework. Finally, Section 6 discusses works related to data imputation for POI recommendation, while Section 7

summarizes the main conclusions and future directions, including a discussion about the main limitations of this work.

2 BACKGROUND

2.1 Point-of-Interest recommendation

Recommender Systems are powerful tools designed to navigate through vast on-line databases and product catalogs. Their primary goal is to simplify the user's search process by filtering out irrelevant data and suggesting items that align with the user's specific preferences. As discussed in the introduction, recommending Points-of-Interest (POIs) such as restaurants, museums, or parks shares similarities to traditional recommendation tasks but has unique characteristics. While typical recommendation scenarios often deal with single-use items, POIs are often revisited, offering valuable data for recommendation systems. Additionally, POI recommendations are influenced by spatial factors (e.g., location), social dynamics (e.g., the user's social network), and temporal aspects (e.g., time of visit).

As a result, many strategies developed for traditional RSs, such as similarity-based methods, matrix factorization, and neural networks, have been adapted to incorporate these additional dimensions. For instance, in [35] and [30], the authors proposed matrix factorization approaches that integrate geographical information, addressing the spatial aspect of POI recommendations. Similarly, in [70], the authors defined a new weighted matrix factorization approach that considers social, temporal, and geographical information. In [29], the authors utilized a Large Language Model (LLM) to process temporal, geographical, and categorical information, while [68] proposed a Recurrent Neural Network (RNN) that incorporates spatio-temporal factors into the recommendation process. Hybrid models that combine various strategies further demonstrate the flexibility of traditional methods when extended to the POI domain. For example, in [63], the authors propose a fusion model combining the user preferences (including temporal information) through a similarity based approaches with a probabilistic approach for modeling the geographical information, while in [57] the authors combined a two-dimensional Gaussian kernel density estimation method with a one-dimensional power law function exploiting the activity of the users in the system.

2.2 Sparsity reduction in recommendation

As discussed in the introduction, higher levels of sparsity are considered one of the biggest challenges for the recommender systems community, and in particular, for POI recommendation, as they entail lower performance values for the algorithms, since the useful signal that can be exploited in a user- or item-basis is very low [34]. Here we shall focus on approaches derived within the recommendation area to address this issue. There are, however, other areas where researchers tackle this problem, as it has an impact on prediction accuracy in the more general area of Machine Learning, mainly by performing *data augmentation* to reduce overfitting and train the models with more data [38], or to increase the power of statistical methods [20]. In this section, we mention several approaches that have the same goal in mind, even when they do not use the terms data augmentation or imputation explicitly.

Significant work has been conducted to reduce sparsity in the context of traditional recommendation, particularly for collaborative filtering (CF). One of the first works devoted to this is [59], where the authors consider a Machine Learning classifier (decision trees, logistic regression, etc.) to decide the missing values, either using the rating matrix or content data, and the results are always better than only using a CF method in terms of error reduction. A similar approach is devised in [64], where the authors exploit users' demographic information by considering simple techniques, such as the average rating of users in the same age range or occupation, to improve the

accuracy of CF algorithms. These two works were focused on neighborhood-based approaches, in [47, 46, 48] the authors present a method with theoretical basis to achieve the maximum imputation benefit, again applied to neighborhood-based CF; the proposed auto-adaptive imputation method and adaptive maximum imputation framework allow adapting to each user's rating history and maximizing the imputation benefit by identifying the imputation area according to the maximum possible item and user sets for each prediction. Results in terms of error reduction are positive, and the authors conclude that what to impute, i.e., deciding *key* missing values, is as important as how it is imputed. Another work tailored for CF based on neighbors is presented in [4], that applied data imputation techniques to identify better neighbors for users, under the hypothesis that a lower sparsity should allow finding more effective neighbors.

Recently, other approaches have been investigated to cover recommendation algorithms beyond those based on neighbors. For example, in [2] the authors propose an imputation method that works for Nonnegative Matrix Factorization, by first imputing all missing values by considering the trust network of users, and then factorizing the original user-item matrix and the imputed matrix, performing predictions through a combination of imputed and non-imputed factorized matrices. This method, still evaluated according to error reduction, obtains positive results, in particular for cold-start and new users. A new matrix factorization technique is proposed in [45], where an imputed matrix is introduced in the cost function.

Other works in recent years exploit deep learning approaches, such as [28], where the authors used a variational autoencoder to extract the features of users and items, under the hypothesis that pre-use preferences (impressions) on items lead to their post-use preferences (ratings); this approach is agnostic to the CF algorithm and shows consistent positive results in terms of accuracy ranking metrics. A different idea is proposed in [25], where the authors aim to quantify the uncertainty on each missing entry and those with the lowest uncertainty are imputed; this idea is implemented on top of a variational autoencoder model, outperforming other simple imputation approaches at different imputation rates in terms of ranking accuracy, although the authors note that performance drops if too many ratings are imputed.

Data sparsity is also a great challenge for sequential recommendation, that is why in [62] the authors augment the dataset by employing counterfactual inference techniques, which are guided by user feedback through reinforcement learning. In [18], the authors propose different operators to augment data in sequential recommendation by transforming non-uniform sequences to uniform ones. In the context of POI recommendation, [33] and [32] propose methods to increase the missing check-ins and boost the performance of next-POI recommendation models (these approaches will be further discussed in the related work section).

In recommendation, since users interact with the system, one possibility not available in other Machine Learning applications is to ask users to rate some items, as a form of rating or *preference elicitation*, or more generally, active learning [56, 19]. This is especially useful for cold-start users, i.e. those users with none or very few ratings collected in the system. The goal of these techniques remains the same as those already discussed: reducing data sparsity by collecting more information, however, they require some type of interaction with users. Moreover, in [27] the authors exploit the concept of *uninteresting items*, which are detected on a user basis and then included (i.e. imputed) as negative feedback for that user. Although it is a very limited imputation technique, it has demonstrated good results in terms of precision when dealing with implicit feedback. Additionally, when privacy is a concern, recent works aim to perturb the information in the system to confuse potential attackers [39]; some of these *obfuscating techniques* may add or remove ratings – when adding ratings these works would enter the same category as data augmentation techniques, however, since the final goal is not reducing the sparsity, they may not have a comparable utility as those techniques already discussed.

3 AN IMPUTATION FRAMEWORK FOR POINT-OF-INTEREST RECOMMENDATION

When imputing new data to increase the information available in the training set, we will end up obtaining a new training set that contains all the original information plus extended information. This can be expressed as follows:

$$C' = \mathcal{F}_{\mathcal{I}}(C, \theta_c, \theta_v) \tag{1}$$

where C' denotes the imputed check-in matrix, and C is the original frequency matrix obtained from the training set. Please, note that C' includes the information from C and the imputed (new) values according to an imputation function represented with $\mathcal{F}_{\mathcal{I}}$. Although in this definition we imply a check-in matrix is needed, this framework can be used in the more traditional recommendation domain by plugging in a rating (or interaction) matrix $\mathcal{R}$.

According to some previous works [47, 28, 25], the problem of preference imputation can be divided into two different parts: a) selecting which preferences should be imputed, and b) how to determine the imputed values. Hence, to address **RQ1** (*What are the required building blocks for an imputation framework tailored for Point-of-Interest recommendation?*) in Equation 1 we use θ_c and θ_v to denote the parameters encoding these characteristics. Hence, θ_c will be used to represent the parameters that model *which check-ins should be imputed*. For example, the number of check-ins to impute, the specific users/venues affected by the imputation, or the specific venues to impute. On the other hand, θ_v will refer to the *mechanism to generate the imputed values*, or the set of values used for imputation. In addition, when adapting this problem to the Point-of-Interest recommendation scenario, ratings, which are usually the basis for imputation in classical recommender systems, are not normally available [65]. Nevertheless, as presented in Section 1, other information sources, such as geographical and temporal data, have proven to be influential in this domain and can be exploited. Therefore, we show below the following parameters encapsulated in the aforementioned $\theta = \{\theta_c, \theta_v\}$ variables:

- Parameters related to θ_c (which check-ins should be imputed):
  - Δ: the number or percentage of check-ins to impute. This can be applied globally across the entire dataset, or on a per-user basis.
  - $\mathcal{U}_{\mathcal{F}_{\mathcal{I}}}$, $\mathcal{I}_{\mathcal{F}_{\mathcal{I}}}$: the specific users or venues affected by the imputation. We could consider only a specific profile of users or items that may suffer the imputation (e.g., cold-start users, less popular items, etc.).
  - $\mathcal{I}_{u,\mathcal{F}_{\mathcal{I}}}$: the specific POIs to impute for a specific user. Here, we can consider domain-specific restrictions. For example, we can incorporate geographical information by only considering items for imputation if they are close to the current or typical location of their corresponding user.
- Parameters related to θ_v (mechanism to generate the imputed values):
  - $\mathcal{A}$: the algorithm/mechanism to generate the imputed check-ins considering the selected parameters from θ_c. In general, regardless of the algorithm used to impute check-ins, only those that maximize the utility of user u for venue i will generally be imputed. In some cases, a specific threshold α can be considered to determine whether the new check-in is of sufficient quality or confidence to be imputed.
  - $\mathcal{V}$: the set of value(s) used for imputation. These values could be predefined by always imputing the same value, based on the user and venues characteristics, or depending on the output of the algorithm or mechanism $\mathcal{A}$ (for example, by exploiting its uncertainty).

It is worthwhile to note that, even though these parameters can (and will later, in our experiments, be) defined manually according to some criteria or use cases in mind, it remains possible to benefit from an automatic decision process on top, so that those parameter values optimizing whatever pre-defined goal observed in the data, could be used in subsequent instantiations of the framework.

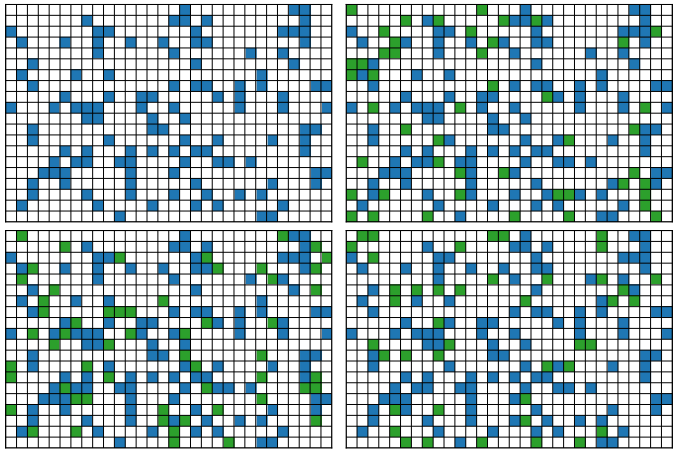

Fig. 1. Representation of the effect of different imputation mechanisms on an original interaction matrix presented in the top-left subfigure. The top-right subfigure shows the same matrix after a 40% increase in the original data, with the imputed data highlighted in green. The bottom-left subfigure represents an imputation strategy where each user's data is increased by 40%. At the bottom-right subfigure another imputation mechanism is depicted, where 40% of the total interactions are added by specifically increasing the data for users with fewer interactions.

This would be related to recent advances in AutoML [21, 55] or its applications to recommender systems [69], and would result in a promising research line to improve the proposed framework in the future.

Toy examples

To illustrate the behavior and variability of this framework, we now present Figures 1 and 2. First, in Figure 1, we show four subplots to capture the effect of some parameters related to θ_c, representing different interaction matrices, where each row corresponds to a user and each column to an item (or venue, in this context). Blue cells indicate that user u has interacted with item i, while white cells indicate that there is no interaction. The top-left subplot shows the original interaction matrix, representing the observable data in a given dataset. When generating new interactions to impute into the original dataset, various strategies and imputation percentages need to be considered, encapsulated in our variables Δ, $\mathcal{U}_{\mathcal{F}_\mathcal{I}}$, and $\mathcal{I}_{\mathcal{F}_\mathcal{I}}$. For instance, we could increase the total number of system interactions by 40%. This scenario is depicted in the top-right subplot. Alternatively, we could apply a 40% increase to each user's interactions but on a user basis, not overall, as shown in the bottom-right subplot. Another approach could be to target only certain users for imputation, such as those with few interactions, which is represented in the bottom-left subplot. As demonstrated, with only two parameters of θ_c, we can generate a wide range of configurations. Here, we have only presented a few of the more straightforward examples. Note that the purpose of this figure is to serve as an example of how an interaction matrix would behave as the data increases. In a real environment, increasing the data by 40% could be counterproductive, as some of the imputed data might lack sufficient quality and result in unrealistic interactions. Moreover, imputing too much data may not only degrade recommendation quality, but also increase computational costs, resulting in models that scale poorly and require longer training times.

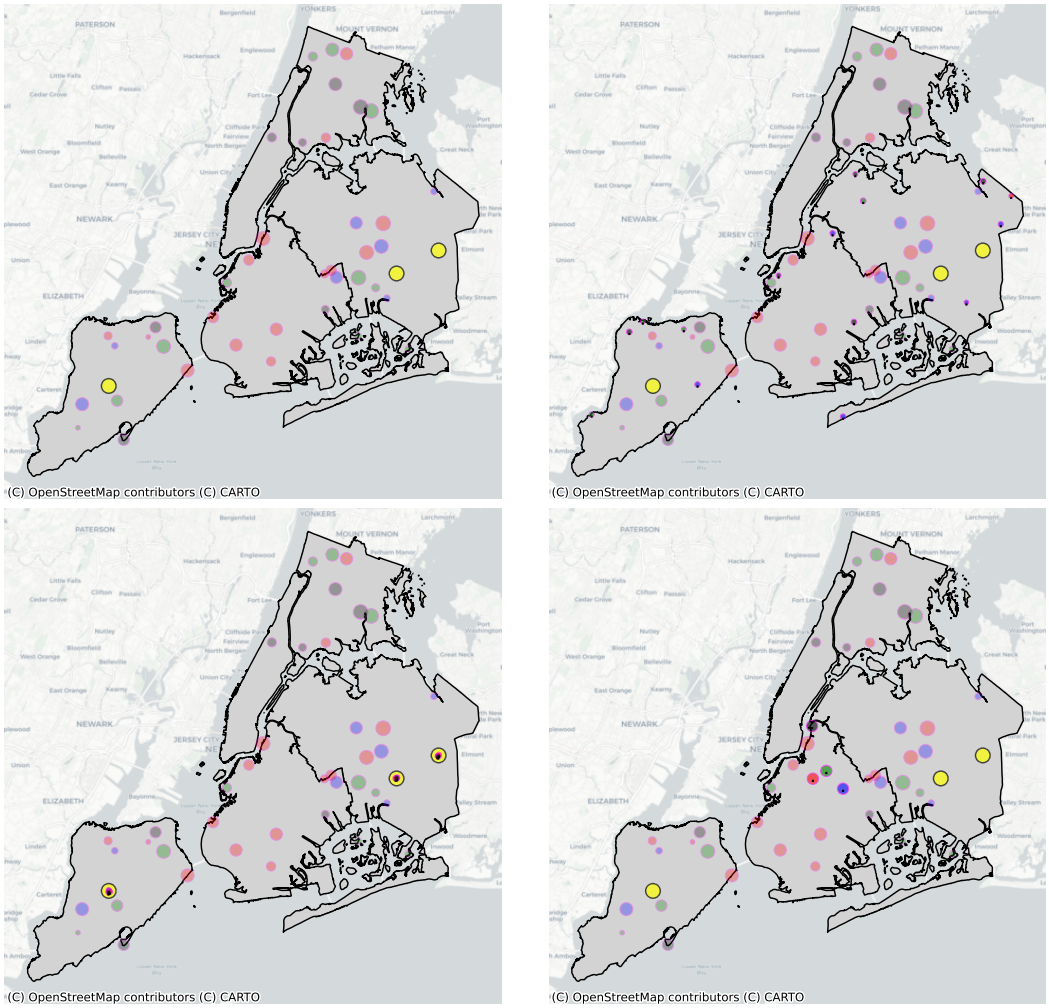

Fig. 2. Representation of the effect of different imputation algorithms $\mathcal{A}$ in New York. The original check-ins are shown in the top-left subfigure, where each color except yellow represent a different user; popular venues are represented in yellow and new imputed interactions contain a small black dot. Top-right shows the effect of imputing check-ins randomly. Bottom-left shows imputation based on popularity. Bottom-right shows imputed interactions near the midpoint of the user.

Now, to demonstrate the effect of some of the parameters encoded by θ_v, Figure 2 shows the effect of applying different algorithms to generate new interactions. The original check-ins for New York are shown in the top-left subfigure, where each color except yellow represents a different user; the yellow points represent popular venues. In the top-right subfigure, we show the effect of imputing some check-ins randomly, where the new imputed interactions contain a small black dot. In the bottom-left subfigure, we show the effect of performing imputation based on popularity, with new imputed values belonging to the set of popular venues, and in the bottom-right, we show the effect of imputing interactions near the midpoint of the user. Note how the result of the imputation mechanism is very different depending on the algorithm considered, either promoting

points closer to where the user already had exhibited some preferences for (last example), popular items (bottom-left), or scattered throughout the city (randomly).

Analysis on computational complexity

Our proposed imputation framework is aimed at improving the effectiveness of the recommendation model it is applied to. However, imputing new data may incur in additional complexity or reduced efficiency on the overall system. We analyze these concerns in this section.

On the one hand, among the parameters identified in our framework, those related to which check-ins to impute, θ_c, could impact significantly in the increased complexity of the model, since they could require to impute too many new check-ins. Hence, controlling these values depending on the existing information and the desired sparsity levels, could make the imputation process to be manageable and useful, or very expensive and probably noisy if too many check-ins are requested but too little information is known in a user basis. On the other hand, the parameters related to the imputation mechanism, θ_v, may bring scalability or memory issues depending on the actual algorithm used to impute the check-ins. Some methods may not scale well to very large user-item matrices, and the imputed matrices, as they are less sparse, can require significant memory resources.

However, such computational costs and efficiency concerns can be mitigated by first noting that imputation occurs **before training**, so the extra time does not impact system's response or its scalability in practical applications, although it may increase the time needed to train the algorithms. Second, some of the components described before and included in our framework can be **parallelized** to improve efficiency. And third, by reducing data sparsity, imputation can lead to **more accurate** recommendations; this can translate to better user engagement and satisfaction, which can indirectly improve efficiency in terms of user retention and conversion. In particular, this means that, even when the original data is large, the imputation process might not *necessarily* be slow or inefficient, as some of the aforementioned imputation parameters may not depend on the size of the dataset; for example, when the imputation mechanism $\mathcal{A}$ does not depend on the number of users or items in the collection.

4 EXPERIMENTS

In this section, we describe the details of the experiments conducted in our study. First, in Section 4.1 we describe the dataset used in the experiments. Then, in Section 4.2 we explain the procedure to perform the training and test splits together with the set of algorithms used in the experiments. Finally, in Section 4.3 we further develop the methodology considered.

4.1 Dataset

In this paper, we have used the global-scale check-in dataset from Foursquare [66], the Gowalla dataset from [13], and a Yelp dataset[1]. The Foursquare dataset contains more than 33M check-ins from 415 cities in 77 countries over more than 2 years. We decided to focus on the cities of London (LON), New York City (NYC), Santiago (SAN), Mexico City (MEX), Moscow (MOS), and Tokyo (TOK) as these cities entail different ratios of users, venues, and check-ins, and each city belongs to a different continent and cultural background. The Gowalla dataset contains 6,442,892 check-ins. In this case, we decided to focus on the following two cities: San Francisco-Gwl (SF-G) and New York-Gwl (NY-G). For Yelp, unlike the other datasets, we use it as a whole and do not divide between different cities[2], so we work with the original 216,471 check-ins. Our motivation

[1] https://github.com/rahmanidashti/LBSNDatasets/tree/master/Yelp/Yelp_4

[2] Note that most of the check-ins in this dataset belong to cities of the United States of America.

Table 1. Statistics of the cities used in the experiments. We present the number of users (|**U**|), the number of items/venues (|**V**|), the number of check-ins (|**C**|), and the sparsity $\delta = (|\mathbf{U}| \cdot |\mathbf{V}|) / |\mathbf{C}|$ for the complete, training, and test splits. The cities from Foursquare dataset (all except Yelp, SF-G, and NY-G) are ordered by the number of check-ins in the complete set from lowest to highest.

City	Split	|U|	|V|	|C|	δ
London (LON)	Complete	7,574	16,554	100,084	0.0798%
	Training	6,492	15,185	80,067	0.0812%
	Test	3,211	7,934	20,017	0.0786%
New York City (NYC)	Complete	12,480	25,082	190,725	0.0609%
	Training	11,097	22,527	152,580	0.0610%
	Test	5,930	12,752	38,145	0.0504%
Santiago (SAN)	Complete	7,415	39,095	289,418	0.0998%
	Training	9,047	47,763	305,732	0.0708%
	Test	6,025	24,133	76,433	0.0526%
Mexico City (MEX)	Complete	9,520	47,213	376,663	0.0838%
	Training	7,220	33,913	231,534	0.0946%
	Test	5,209	17,814	57,884	0.0624%
Moscow (MOS)	Complete	9,534	55,156	382,165	0.0727%
	Training	9,188	41,320	301,330	0.0794%
	Test	6,744	23,099	75,333	0.0484%
Tokyo (TOK)	Complete	11,594	51,467	441,625	0.0740%
	Training	11,077	47,608	353,300	0.0670%
	Test	7,793	25,458	88,325	0.0445%
Yelp (Yelp)	Complete	5,000	13,683	202,566	0.2960%
	Training	4,599	12,984	162,052	0.2714%
	Test	3,181	9,073	40,514	0.1404%
New York-Gwl (NY-G)	Complete	5,671	15,761	97,692	0.1093%
	Training	4,918	14,688	78,153	0.1082%
	Test	1,685	8,740	19,539	0.1327%
San Francisco-Gwl (SF-G)	Complete	6,113	14,847	134,442	0,1481%
	Training	5,195	13,999	107,553	0,1326%
	Test	2,217	9,157	26,889	0,1325%

behind this decision was to evaluate the performance of our framework in a different setting that is sometimes considered in the literature [53].

4.2 Experimental Setup

As a first step regarding the data preparation, we conducted a pre-processing step for the entire Yelp dataset and for every city in both Foursquare and Gowalla datasets (that is, we considered each city as an independent dataset). This involved applying a 2-core filtering process, where we maintained only those users and venues with at least 2 check-ins. To handle repeated visits by the same user to the same POI, we aggregated these visits creating a frequency matrix, preserving the timestamp of the last check-in.

Hence, for each independent dataset, we then split the data into training and test subsets using a global temporal split. Specifically, we allocated the oldest 80% of all aggregated check-ins for training, while the remaining 20% were used to test the performance of the approaches. We present in Table 1 the specific details of the cities, where we report the statistics of the complete city, the training, and test sets. Note that the information of the test set is unknown to the models, as they only work with the data from the training file.

Following this, we selected a set of traditional top-n recommenders. Our selection included both classical recommendation models, which operate independently of geographical factors, and Point-of-Interest recommenders, which incorporate geographical components. For each dataset, we performed hyperparameter tuning on each model (see Table 4, in Appendix A), selecting the best configurations based on their performance in terms of nDCG@10.

Furthermore, there are different strategies to consider when selecting the candidate items (venues) for recommendation. In this work, we use the TrainItems strategy. This approach considers, as candidate items for recommendation to each user, all the POIs in the training set the user has not visited during the training step [51]. We have decided to work with this strategy because the general purpose of a recommender should be to suggest venues to users they have not visited before [42].

Below, we present the classical and Point-of-Interest recommenders used in the experiments to analyze whether our imputation framework is capable of improving their performance.

- Rnd: recommends random Points-of-Interest to the users.
- Pop: recommends the Points-of-Interest that have received the largest number of visits from different users.
- UB and IB: non normalized versions of collaborative filtering techniques based on user and item neighborhood, respectively [42].
- HKV: matrix factorization (MF) algorithm that uses Alternate Least Squares for optimization [24].
- BPR: Bayesian Personalized Ranking (BPR) loss from [49] applied to matrix factorization.
- EASEr: Embarrassingly Shallow Autoencoders for Sparse Data proposed in [58].
- SAE-NAD: neural network approach from [37] based on autoencoders. It uses a self-attentive encoder (SAE) (to learn the user-POIs relationships) and a neighbor-aware decoder (NAD) to model the geographical influence.
- IRenMF: weighted MF from [35]. It also models the geographical influence between neighbor POIs.
- GeoBPR: geographical BPR. It incorporates the geographical component into BPR, assuming that users will prefer to visit Points-of-Interest close to the previous visited venues [67].
- FMFMGM: Hybrid technique from [11] combining a Probabilistic MF (PMF) and a Multi-center Gaussian Model approach (MGM).
- RnkGeoFM: MF based on BPR that models the geographical influence using the geographical neighbor Points-of-Interest with respect to the target Point-of-Interest with an additional latent matrix for the users [30].
- PGN: hybrid Point-of-Interest recommendation technique that combines the Pop, UB, and AvgDis recommenders [54]. The AvgDis recommender is a geographical algorithm that recommends the closest Points-of-Interest to the user's average location. The average location is computed by calculating the midpoint of the coordinates of the visited Points-of-Interest in the user profile (i.e., her training set). Since AvgDis on its own is a very basic baseline, we do not report its results, as its accuracy performance is similar to that of a random model.

4.3 Methodology

Producing comparable and reproducible experimental environments is not easy, but it is something the community is investing a lot of efforts in last years [5, 16]. Because of that, as one of our contributions, in this section we present in detail the procedure followed in the experiments, summarized in Figure 3. Thus, for each dataset (independent cities or entire data, depending on the source) used in the experiments, we firstly divide the original dataset in training and test sets. From the training set, we obtain the original check-in matrix C, and we train the recommendation algorithms listed in Section 4.2 using the hyperparameters stated in Table 4 and this (non-imputed) check-in matrix C. Once we have selected the best recommenders according to the optimal hyperparameters by checking the performance against the test set, we use these optimal parameters to report the performance of the non-imputed versions of the algorithms.

In parallel, we generate imputed check-in matrices using different combinations of parameters θ_c and θ_v, as explained in Section 3. For each imputed matrix C', we train again every recommendation algorithm but using the optimal parameters found for the non-imputed training data. It is important to include this additional training step with the imputed check-in matrix C', as the model can potentially learn new patterns from the imputed data, even without further parameter tuning. Finally, using again the original (non-imputed) test set, we evaluate these recommenders trained with the extended training set.

It is important to note that the test set remains the same throughout the entire process. Moreover, since imputed check-ins may belong to the test set and typical recommendation algorithms do not

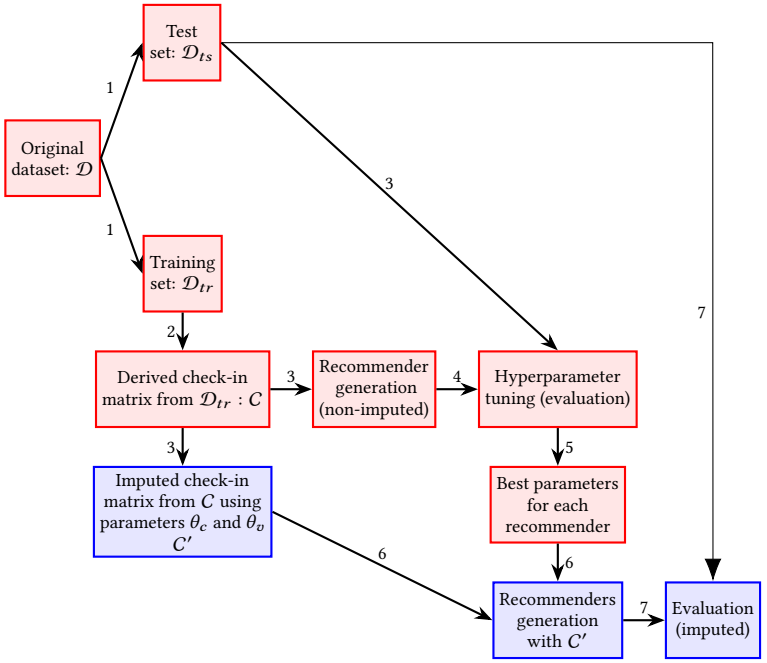

Fig. 3. Summary of the methodology followed in the paper. The numbers along the arrows indicate the order in which the steps should be executed, with each step depending on the successful completion of all preceding steps with lower numbers. In red, we represent the steps that use the original training check-in matrix C. In blue, we represent the steps involving the imputed check-in matrix C'.

expect repetitions between training and test sets, to ensure a comparable environment between imputed and non-imputed scenarios, we force every algorithm to follow the TrainItems strategy (see Section 4.2) but according to the original matrix, i.e., all the algorithms will consider the same set of candidate items, irrespectively of whether that item already existed in its training set (because it was imputed) or not.

Also, it is worth highlighting that optimizing the hyperparameters on the imputed algorithms would probably obtain better results, but would demand a huge computationally overload while not being very realistic. Indeed, we argue the use case scenario of such an imputation framework for recommendation should work *on top* of previously trained algorithms, which are already optimized, with the goal of producing better results just by performing *one additional training* with the imputed input, but not to start from scratch in every case.

5 RESULTS

In this section, we present the effect of applying our imputation framework with different configurations and analyze the performance of the recommenders after augmenting the data. Before that, to provide a context to these results and to get performance baselines (i.e., the recommenders trained without the imputation matrix) to compare against, we show the results obtained by the models in the traditional POI recommendation scenario; that is, without applying our imputation framework, i.e., working only with C. All experiments report the results of the models in terms of ranking accuracy (nDCG) and novelty and diversity. Novelty is measured using the Expected Popularity Complement (EPC) metric and diversity with Gini [8]. In all cases, the higher the value

Table 2. Results of evaluated recommenders across different datasets and metrics. For each dataset, we show the performance of each of the recommenders presented in Section 4.2 in terms of ranking accuracy (nDCG), novelty (EPC), and diversity (Gini). All metrics @10. Best result in a row (specific metric for a particular dataset) in bold.

City	Metric	Rnd	Pop	UB	IB	HKV	BPR	EASEr	SAE-NAD	IRENMF	GeoBPR	FMFMGM	RnkGeoFM	PGN
LON	nDCG	0.0001	0.0643	0.0569	0.0258	0.0362	0.0639	0.0769	0.0752	**0.0936**	0.0727	0.0550	0.0438	0.0691
	EPC	**0.9992**	0.9385	0.9720	0.9931	0.9867	0.9396	0.9553	0.9448	0.9510	0.9640	0.9574	0.9711	0.9443
	Gini	**0.5636**	0.0006	0.0233	0.2578	0.0214	0.0006	0.0063	0.0013	0.0060	0.0073	0.0033	0.0873	0.0060
NYC	nDCG	0.0002	**0.1474**	0.1015	0.0346	0.0870	0.1380	0.1290	0.1377	0.1414	0.0960	0.0563	0.0724	0.1398
	EPC	**0.9994**	0.9076	0.9508	0.9876	0.9578	0.9141	0.9339	0.9205	0.9138	0.9660	0.9405	0.9560	0.9268
	Gini	**0.5868**	0.0004	0.0230	0.2564	0.0067	0.0014	0.0051	0.0013	0.0017	0.0006	0.0011	0.0525	0.0434
SAN	nDCG	0.0001	0.0797	0.0761	0.0349	0.0395	0.0735	0.0797	0.0818	**0.1031**	0.0639	0.0322	0.0449	0.0881
	EPC	**0.9991**	0.8653	0.9535	0.9692	0.9599	0.8721	0.9109	0.8869	0.8892	0.9266	0.9114	0.9368	0.9112
	Gini	**0.4887**	0.0003	0.0140	0.1794	0.0069	0.0004	0.0026	0.0008	0.0024	0.0014	0.0011	0.0070	0.0439
MEX	nDCG	0.0001	0.0581	0.0635	0.0375	0.0433	0.0626	0.0682	0.0758	**0.0910**	0.0583	0.0359	0.0458	0.0701
	EPC	**0.9992**	0.8791	0.9650	0.9772	0.9652	0.8811	0.9276	0.9080	0.9071	0.9398	0.9215	0.9542	0.9117
	Gini	**0.4946**	0.0002	0.0151	0.2113	0.0080	0.0002	0.0026	0.0010	0.0018	0.0014	0.0015	0.0108	0.0320
MOS	nDCG	0.0000	0.0332	0.0521	0.0380	0.0292	0.0348	0.0596	0.0542	**0.0675**	0.0488	0.0316	0.0461	0.0457
	EPC	**0.9993**	0.8811	0.9651	0.9795	0.9696	0.9466	0.9301	0.9074	0.9052	0.9428	0.9088	0.9576	0.9116
	Gini	**0.4531**	0.0002	0.0160	0.2080	0.0085	0.0098	0.0036	0.0015	0.0020	0.0034	0.0009	0.0141	0.0257
TOK	nDCG	0.0001	0.1584	0.1834	0.1337	0.1164	0.1743	0.1912	**0.2062**	0.1883	0.1905	0.1791	0.1296	0.1759
	EPC	**0.9994**	0.8523	0.8941	0.9373	0.9242	0.8542	0.8959	0.8750	0.8674	0.8790	0.8761	0.9339	0.8829
	Gini	**0.5256**	0.0002	0.0033	0.1052	0.0008	0.0002	0.0017	0.0011	0.0007	0.0006	0.0005	0.0128	0.0284
Yelp	nDCG	0.0011	0.0147	**0.0529**	0.0344	0.0493	0.0436	0.0522	0.0427	0.0486	0.0489	0.0261	0.0389	0.0403
	EPC	**0.9973**	0.9558	0.9767	0.9885	0.9803	0.9726	0.9809	0.9742	0.9775	0.9727	0.9867	0.9841	0.9695
	Gini	**0.6047**	0.0007	0.0219	0.1077	0.0249	0.0106	0.0452	0.0132	0.0283	0.0074	0.0847	0.0910	0.0185
NY-G	nDCG	0.0003	0.1535	0.1109	0.0281	0.0823	0.1161	0.1064	0.1143	0.1444	0.1005	0.0455	0.0632	**0.1576**
	EPC	**0.9990**	0.9138	0.9524	0.9921	0.9809	0.9196	0.9534	0.9237	0.9382	0.9262	0.9575	0.9738	0.9238
	Gini	**0.4690**	0.0006	0.0172	0.1502	0.0133	0.0012	0.0172	0.0013	0.0144	0.0014	0.0092	0.1077	0.0141
SF-G	nDCG	0.0007	**0.2261**	0.1838	0.0232	0.1025	0.2015	0.1755	0.1968	0.2117	0.2001	0.0749	0.0749	0.2225
	EPC	**0.9985**	0.9133	0.9585	0.9918	0.9734	0.9243	0.9472	0.9283	0.9390	0.9229	0.9561	0.9699	0.9343
	Gini	**0.5367**	0.0007	0.0224	0.1660	0.0039	0.0032	0.0140	0.0030	0.0154	0.0010	0.0081	0.0760	0.0446

of the metric, the higher the performance of the recommender in that dimension. Unless stated otherwise, we report results @10, i.e, for these metrics we only consider the 10 first recommended items to each user.

In Table 2 we show the performance of the recommenders presented in Section 4.2 in terms of nDCG (ranking accuracy), EPC (novelty), and Gini (diversity) for the 9 datasets (6 cities from Foursquare, 2 from Gowalla, and Yelp). In this table, we observe the low results obtained by the recommenders in terms of ranking accuracy primarily due to data sparsity (see Table 1) and the temporal split performed, where both users and venues might be absent in the training set but present in the test set. Consequently, some recommenders may fail to perform recommendations to all users in the test set. Additionally, using the TrainItems strategy results in new relevant venues in the test set being unavailable as candidates for recommendation. Regarding the recommenders, we observe that in many cases, the Pop recommender is quite competitive, and in the case of NYC and SF-G, it is the algorithm that obtains the best results. This is a common result in the area of POI recommendation, where most of the check-ins are concentrated in a few venues [54]. Interestingly, the Pop model performs relatively poorly on Yelp. This may be due to the sparsity of all datasets; however, Yelp is the least sparse dataset, allowing other recommender models to capture more complex patterns. Nevertheless, focusing only on accuracy presents its own challenges. Although Pop is competitive in ranking accuracy, by definition, it is the worst in terms of novelty and diversity. On the other hand, we can observe the opposite scenario: Rnd performs the worst in terms of ranking accuracy but quite well in novelty and diversity. Regarding the rest of the classical recommenders, UB and BPR also show competitive results when analyzing ranking accuracy. However, a disadvantage of BPR (along with its geographical version, i.e., GeoBPR) is

Table 3. Average percentage of user check-ins by distance (1km, 5km, 10km, 15km, 20km, and higher than 20km) from their corresponding midpoint across the analyzed datasets.

City	1km	5km	10km	15km	20km	>20km
London	24.60%	78.30%	93.40%	97.80%	99.49%	100.00%
New York City	29.67%	77.18%	92.31%	97.77%	99.86%	100.00%
Santiago	11.42%	63.14%	90.00%	97.58%	99.68%	100.00%
Mexico City	10.04%	57.82%	89.60%	98.46%	99.82%	100.00%
Moscow	10.12%	60.56%	88.45%	97.57%	99.64%	100.00%
Tokyo	10.97%	69.95%	93.96%	98.40%	99.66%	100.00%
Yelp	7.95%	40.52%	57.82%	65.63%	69.55%	100.00%
New York-Gwl	27.15%	75.28%	90.60%	96.31%	99.30%	100.00%
San Francisco-Gwl	21.28%	63.38%	83.42%	93.66%	98.57%	100.00%

that it exhibits a strong popularity bias, as evidenced by its diversity results being almost identical to those of the Pop recommender.

However, as expected, considering the geographical component is crucial for making relevant recommendations in this domain. In 5 out of 6 cities of the Foursquare dataset, the top model is a POI recommender (IRenMF for London, Santiago, Mexico City, and Moscow, and SAE-NAD for Tokyo). These models generally perform similarly in terms of ranking accuracy, novelty, and diversity. For the rest of the recommenders, while GeoBPR achieves slightly higher novelty and diversity than BPR, it still scores low overall, especially in the latter. PGN, though not the best in ranking accuracy (except in New York-Gwl), excels in diversity except in London, due to its combination of Pop, UB, and AvgDis, which balances popularity bias with geographical proximity. Given the challenge of balancing these complementary dimensions, the IB recommender is also noteworthy: despite its lower ranking accuracy, it obtains very good results in novelty and diversity, being competitive against Rnd.

5.1 Data imputation applied to all users

Now, we analyze the effect of applying the imputation framework we defined in Section 3 with distinct configurations of parameters θ_c and θ_v to the previously presented recommendation algorithms and cities. Firstly, we focus on the total set of all users in the training set. In this sense, we will impute new check-ins to each training user proportionally to the number of check-ins they have. In other words, we will impute more check-ins to those users who have made a high number of interactions in the training set, in order to encourage collaborative information. Hence, the instantiation of the framework parameters in this experiment would be defined as follows:

- Δ: the percentage of interactions to impute will stay low, as stated in previous literature that too much imputation may add noise [48, 25]. In the experiments, we will compare the impact of using a 10% and 30% imputation ratio, i.e., $\Delta = \{10\%, 30\%\}$. We believe these values would help improve the performance of the recommenders without a loss in the quality of the dataset.
- $\mathcal{U}_{\mathcal{F}_I}, \mathcal{I}_{\mathcal{F}_I}$: all users in the training set will experience an increase in their number of training check-ins. The imputed check-ins will be from Points-of-Interest that appear in the training set, i.e., $\mathcal{U}_{\mathcal{F}_I} = U_{tr}, \mathcal{I}_{\mathcal{F}_I} = I_{tr}$, where U_{tr} and I_{tr} represent the users and items in the training set.
- $\mathcal{I}_{u,\mathcal{F}_I}$: the new imputed check-ins will correspond to venues not visited previously in the training set by the user while being at most 10km away from the user midpoint, i.e., for each

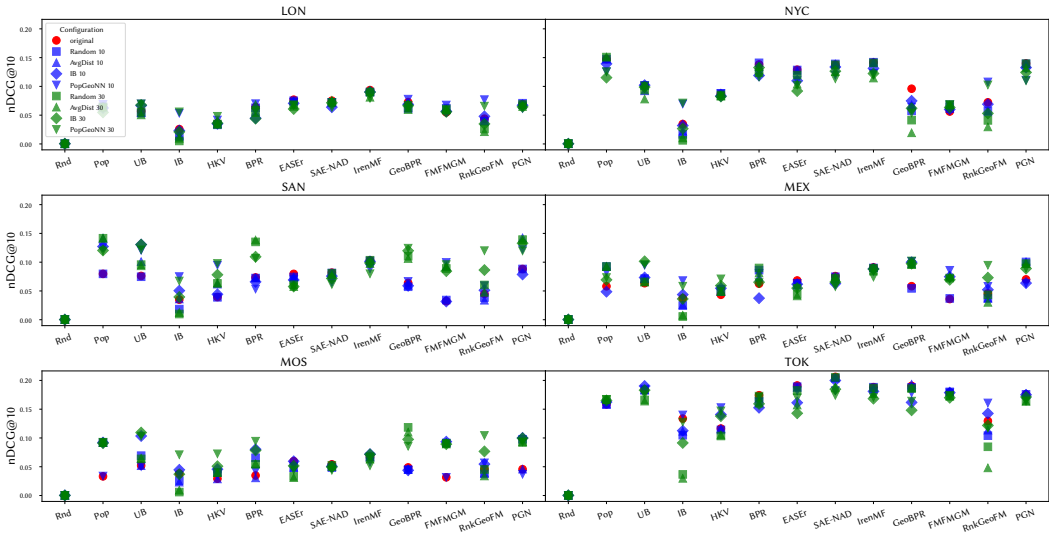

Fig. 4. Performance of each recommender in every Foursquare city for the nDCG@10 metric when imputation is done to every training user. In red, the performance of the model in the original training. In blue, the effect of our imputation techniques, increasing the data for each user by 10%, and in green, by 30%. Each marker denotes a different mechanism to generate the imputed data; i.e., *Rnd 10* denotes that the Rnd algorithm is used to generate 10% imputed check-ins in every user.

user u, $\mathcal{I}_{u,\mathcal{F}_\mathcal{I}} = \{i \in I_{tr} \wedge \text{dist(i, Midpoint}(u)) < 10km \mid i \notin I_{tr_u}\}$), where I_{tr_u} denotes the venues that have been visited by the user in the training set. We use the threshold of 10km because, in all cities, at this distance the average percentage of user check-ins from their corresponding midpoint is higher than 83% (see Table 3 for more details), except for Yelp, which, as previously mentioned, does not have check-ins specific to a single city.

- $\mathcal{A}$: we will use the algorithms Rnd, AvgDis, IB, and PGN for imputing the data, i.e., $\mathcal{A}$ = {Rnd, AvgDis, IB, PGN}, as representative simple methods for geographical and collaborative approaches. In all cases, we will impute check-ins in order of highest to lowest relevance until we get the desired percentage of increased check-ins per user. For IB, we used SetJ (Jaccard) similarity with 90 neighbors and for PGN, we used the same similarity with 100 neighbors. We used these values because they were the parameters that generally obtained the best results when we used them in the recommenders without imputation.
- $\mathcal{V}$: we will use a value of 1 for the imputed venues, i.e., $\mathcal{V}$ = {1} as we assume the user visits the imputed POI only once.

We present in Figure 4 the results obtained in each city of the original global check-in Foursquare dataset using the aforementioned configuration in terms of nDCG@10. We represent with different colors and markers the different configurations. Hence, in red, we represent the value obtained by the recommender using the original dataset (the value of nDCG@10 obtained in Table 2). In blue, we represent the performance of the recommenders increasing the training set for each user by a 10%, and in green, the same but by a 30%. The square, triangle, diamond, and inverted triangle markers represent the imputation performed by the Rnd, AvgDis, IB, and PGN recommenders. We also report the results of the same recommenders for Yelp and Gowalla in Figures 5 and 6.

Analyzing the results to address **RQ2** (*By applying this imputation framework, is it possible to improve ranking accuracy?*), we notice some interesting behaviors. First, we observe that the Rnd

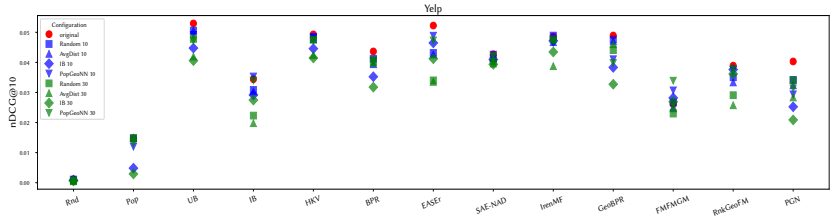

Fig. 5. Recommender performance in Yelp using the same configuration as in Figure 4.

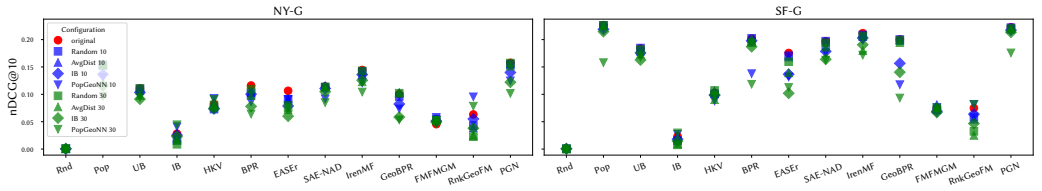

Fig. 6. Recommender performance in both cities from Gowalla (New York City, left and San Francisco, right) using the same configuration as in Figure 4.

recommender obtains practically no performance change through the imputation. This is expected, since there is no personalization component in this method. On the other hand, for most cities in the Foursquare dataset (except for New York City and the GeoBPR recommender and Santiago and Mexico City for EASEr), we identify at least one imputation configuration that leads to improved performance, or, when no improvement is observed, the difference with respect to the original version remains negligible. Recall that the optimal parameter settings for the algorithms were determined using the original training set, not the imputed data set; consequently, if hyperparameter selection were to be performed on the imputed data set, the results could potentially be superior, although less realistic in practice.

Regarding the algorithms used to perform the imputation ($\mathcal{A}$), we observe that, in general, the worst performing one is the Rnd strategy, although in some cases, especially in the case of 30% imputation ratio, we find that this strategy substantially improves the performance with respect to the configuration without imputation, as in the case of the city of Santiago, Moscow, or Mexico City for the Pop, BPR, GeoBPR, or PGN recommenders. Although this may seem strange, we have to take into account that although the imputation is random, we impute only those POIs that are at a distance of at most 10km from the midpoint of each user, so that POIs that are far away are discarded, hence, the imputation is not *completely* random. This again indicates the importance of geographical influence in making recommendations, bringing, thus, a preliminary response to **RQ3** (*Which parameters of this framework are more critical to achieve performance improvements?*), although in the next section we will continue with such analysis.

However, simply recommending nearby venues does not necessarily improve performance, as demonstrated by using AvgDis recommender to impute data, where its performance is sometimes even worse than random recommendations. For instance, in New York City, this is the case with the UB, GeoBPR, and RnkGeoFM methods, and in Moscow, with the BPR and HKV algorithms. In any case, as analyzed previously regarding the results without imputation, both the IB and PGN algorithms are noteworthy for their balance between ranking accuracy, novelty, and diversity. The results here indicate that PGN generally enhances the performance of most models in almost

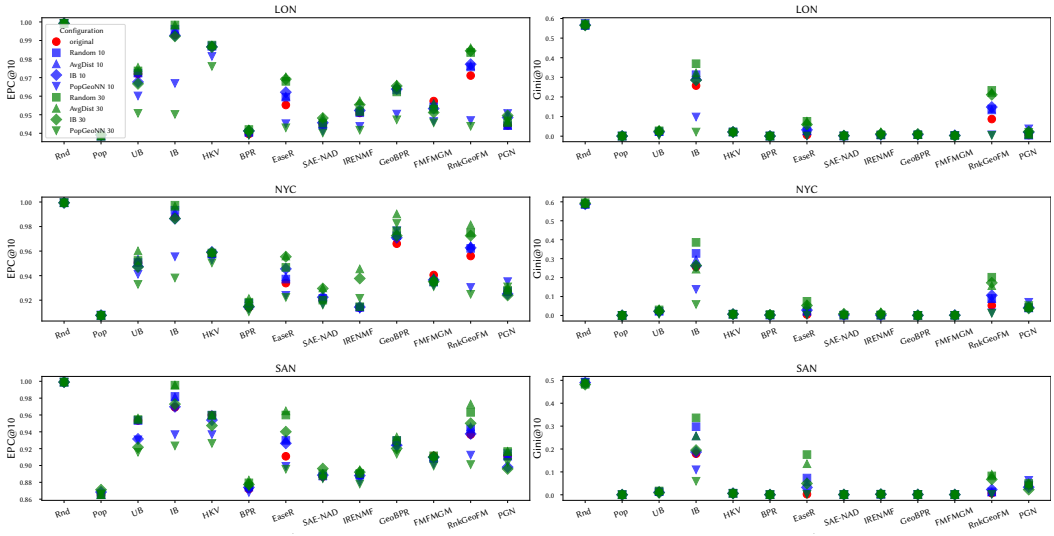

Fig. 7. Figure representing the performance in terms of novelty (EPC) and diversity (Gini), in the Foursquare cities of London (LON), New York City (NYC), and Santiago (SAN) following the same imputation strategy as the one used in Figure 4 (imputing for each training user).

all cities, sometimes yielding remarkable improvements. For instance, in Santiago, both IB and HKV obtain an nDCG performance increase higher than 100%. Similarly, in Moscow, equivalent performance boosts are observed for FMFMGM, RnkGeoFM, HKV, and BPR. In the case of IB, we also observe performance increases for recommenders in several cities, most notably in Santiago and Moscow, and to a lesser extent in Mexico City and Tokyo. However, these improvements are smaller compared to those achieved with PGN. As previously noted in the non-imputed results, IB performance in terms of accuracy was not very competitive.

Let us now analyze the results for Gowalla (Figure 6), where in both cities we observe that the benefits of our imputation strategies are generally smaller than in the case of Foursquare. In neither city we are able to improve the base performance of EASEr, and both BPR in New York-Gwl and IRenMF in San Francisco-Gwl fail to improve with the imputed data. However, in all other cases, at least one imputation technique maintains or enhances performance. Notable improvements include IB and RnkGeoFM in both cities, as well as HKV in New York-Gwl. Regarding the Yelp dataset (Figure 5), we also observe that imputation techniques do not achieve results as strong as in Foursquare, as we do not find any configuration that improves the performance of UB, BPR, EASEr, GeoBPR, RnkGeoFM, or PGN without imputation. While this may seem surprising, it is important to note that the Yelp dataset does not correspond to a single city (see Table 3), making tailored imputations for Points-of-Interest (using geographical and domain-dependent aspects) more challenging in this context. This once again highlights the crucial role of geographic influence in POI recommendation and reinforces the importance of not mixing POIs from different cities [53].

In order to clarify the results obtained in Figures 4, 5, and 6, we present in Appendix B the complementary Table 5. In this table, for each combination of dataset and configuration, we show the number of models that achieve a better result than the imputation-free recommender (original, represented in red in these figures) in terms of nDCG. Based on these summarized results, we observe that the configurations using PGN algorithm are the ones that improve the performance of the algorithms the most, followed by the IB strategy.

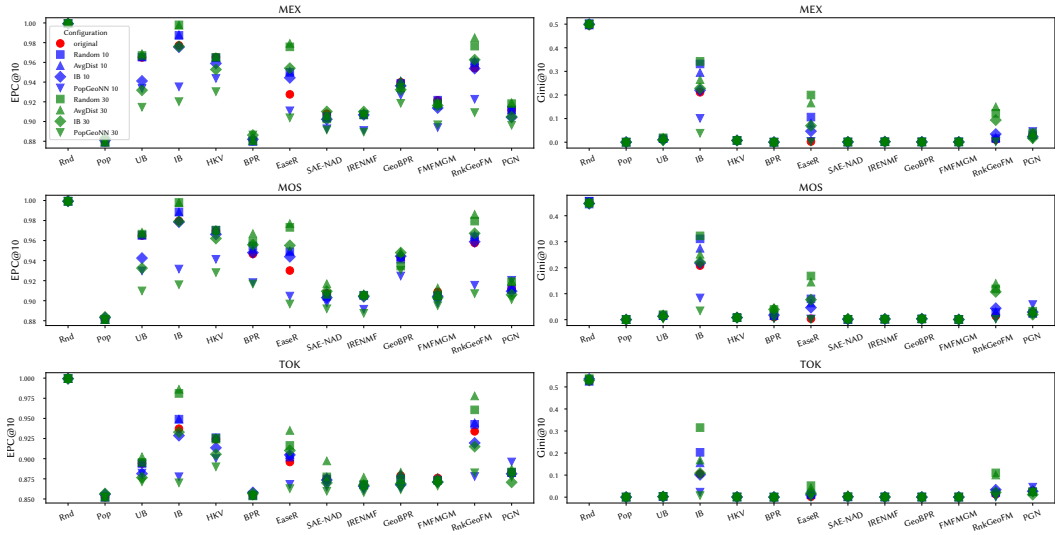

Fig. 8. Figure representing the performance in terms of novelty (EPC) and diversity (Gini), in the Foursquare cities of Mexico City (MEX), Moscow (MOS), and Tokyo (TOK) following the same imputation strategy as the one used in Figure 4 (imputing for each training user).

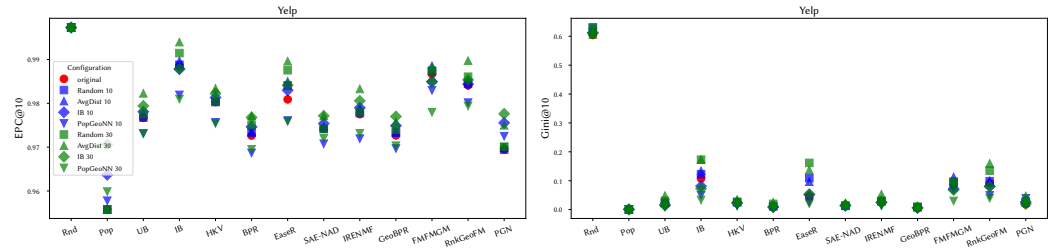

Fig. 9. Novelty (EPC) and diversity (Gini) performance in Yelp as in Figures 7 and 8.

Once we have analyzed the performance of the imputation strategies in terms of ranking accuracy, let us analyze these strategies but in terms of novelty and diversity, in order to answer **RQ2**. For this reason, Figures 7 and 8 present the results for all the cities of the Foursquare dataset using the EPC (novelty) and Gini (diversity) metrics, whereas Figures 9 and 10 show the results for Yelp and Gowalla. In the case of Foursquare, we observe a well-known phenomenon in recommender systems: recommenders that perform poorly in terms of accuracy ranking often excel in novelty and diversity. This highlights the challenge of finding a balance between all dimensions [60].

Nevertheless, it is possible to observe that several recommenders keep consistent (unchanged) performance in these metrics, regardless of the imputation mechanism used. For example, the Pop recommender and BPR show the same results in both novelty and diversity, while IRenMF, FMFMGM, HKV, and UB remain unchanged in terms of diversity. This consistency is not necessarily negative; as demonstrated in the previous section, we were able to improve performance in terms of accuracy ranking. This indicates that there are configurations capable of enhancing relevance without sacrificing performance in novelty and diversity. Moreover, we can also observe some interesting cases, as in the RnkGeoFM recommender, where we improve both novelty and diversity,

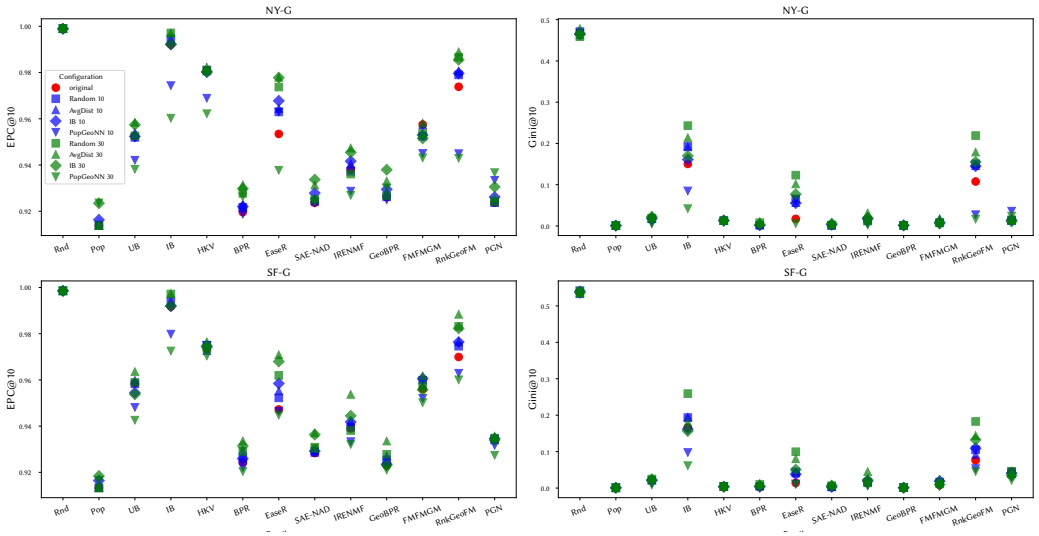

Fig. 10. Novelty (EPC) and diversity (Gini) performance in Gowalla (New York City in top row, San Francisco in bottom row) as in Figures 7 and 8.

while at the same time, as we saw before, we are able to improve the performance of nDCG as well. Additionally, in the case of EASEr, the imputed and non-imputed versions showed almost identical accuracy performance. However, certain configurations exhibited substantial increases in novelty and diversity. For SAE-NAD, novelty and diversity remained largely unchanged between the imputed and non-imputed versions, except for specific cases such as Tokyo or New York City.

In any case, we found that the Rnd and AvgDis strategies generally have a positive impact on the novelty and diversity of the models. However, these strategies do not achieve a significant improvement in terms of ranking accuracy. This highlights the ongoing challenge of balancing enhancements in ranking accuracy while improving novelty and diversity. Nonetheless, as a positive contribution of our work, it also suggests that certain combinations within the proposed framework might be able to enhance all three dimensions simultaneously.

Following the analysis of Foursquare, the novelty and diversity analysis for the cities of Yelp (Figure 9) and Gowalla (Figure 10) evidence that some trends remain consistent with those seen in Foursquare cities. Once again, EASEr, RnkGeoFM, and IB benefit the most in these evaluation dimensions when applying our imputation techniques. Furthermore, diversity appears to be less affected than novelty. In fact, for novelty, in almost all cases (except for the Rnd recommender), we find at least one configuration that outperforms the non-imputed version.

Again, we summarize in Tables 6 and 7 (in Appendix B) the results presented in previous figures, showing the number of models that outperform their non-imputed versions in each combination of dataset and configuration in terms of EPC and Gini.

5.2 Data imputation applied to cold-start users

In the recommender systems domain, it is common to analyze the performance for specific user groups. These groups can be defined by metadata (e.g., age, gender, ethnicity) or by their behavior when interacting with the system. One of the most common groups is cold-start users – those with very few interactions, making it challenging to generate accurate predictions for them. In this

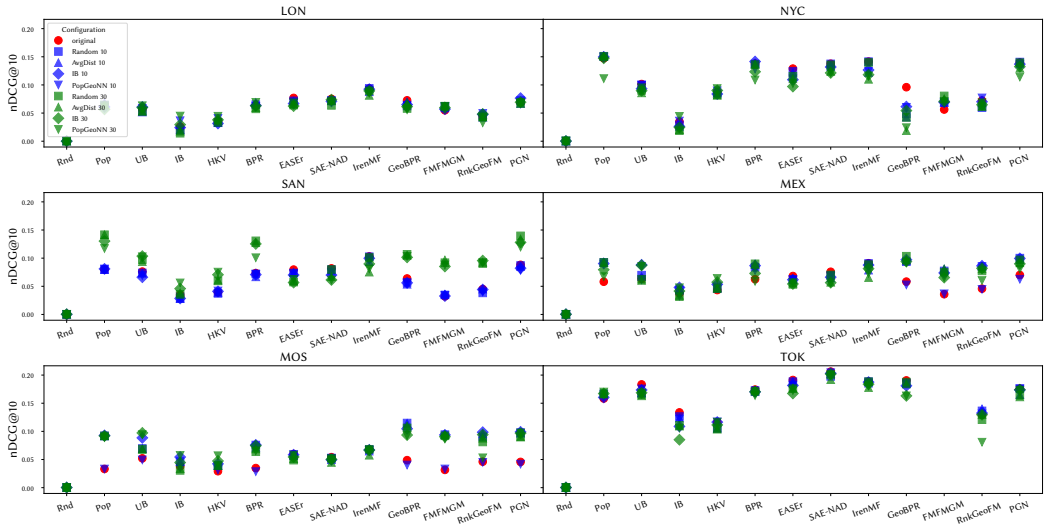

Fig. 11. Same as Figure 4, but imputation is only applied to users with the lowest number of check-ins (cold-start users) as explained in Section 5.2 (Foursquare).

section, we analyze the effect of our imputation techniques on these types of users, as we want to confirm if they are able to benefit from our framework.

In this case, parameters $\mathcal{I}_{u,\mathcal{F}_\mathcal{I}}$, $\mathcal{A}$, and $\mathcal{V}$ remains the same as in the previous experiment. Although we are also using a 10% and a 30% of increment for Δ, in this scenario, the imputation is applied globally rather than per each user. This means the increase is based on the total number of check-ins. More differently, now the users with the lowest number of check-ins (cold-start) are the ones affected by the imputation process ($\mathcal{U}_{\mathcal{F}_\mathcal{I}}$). The procedure involves ordering users by their number of check-ins, computing the average number of check-ins needed to meet the stipulated data increment, and impute check-ins for each user until they reach the computed average. Once a user reaches such average value, their data is no longer imputed, and the process moves to the next user. The procedure stops once the overall imputation target value is achieved. Consequently, users with a higher number of check-ins will not experience an increase in their interactions.

Results for this scenario in the Foursquare, Yelp, and Gowalla datasets are shown in Figures 11, 12, and 13. For Foursquare, when comparing these outcomes with the previous experiment, we see that the behavior is slightly different, as there are more instances where configurations fail to outperform the value obtained from the original training without imputation. While Figure 4 only showed that for the GeoBPR in New York City the non-imputed version is the best, in these experiments it also happens in Tokyo for UB, IB, and BPR, and also in London for GeoBPR. Nevertheless, we also find some instances where results are slightly better than in the previous experiment, such as for Santiago or Moscow with PGN and a 30% increment.

The results for Yelp (Figure 12) and Gowalla (Figure 13) show that, although there are still models that do not benefit from the imputation strategies (e.g., BPR, EASEr, SAE-NAD, and IRenMF in both San Francisco-Gwl and New York City, and UB, PGN, and GeoBPR in Yelp), the performance loss in these cases is generally negligible – except for UB in Yelp. On the other hand, for the rest of the models where performance improves, the gains in terms of nDCG are clearly noticeable.

As in the previous experiment, we also present a summary in the appendix (in Table 8) with the number of models that are able to obtain a higher performance than their non-imputed version

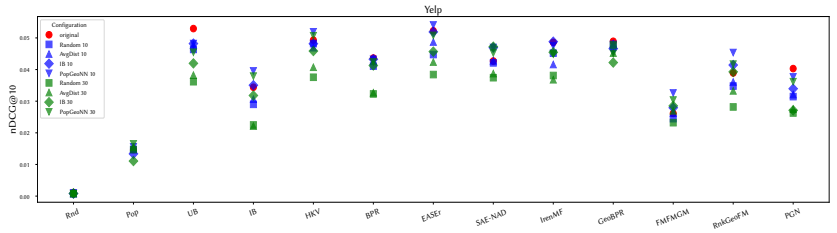

Fig. 12. Same as Figure 5, but imputation is only applied to cold-start users (Yelp).

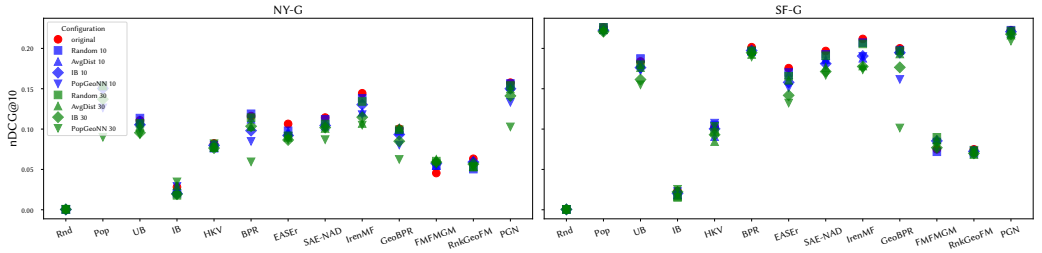

Fig. 13. Same as Figure 6, but imputation is only applied to cold-start users (Gowalla).

for each city and configuration. Despite these results, where more combinations of algorithm and imputation mechanism do not improve performance, imputing only for users with fewer check-ins could still be a valid strategy, as the overall trend is positive. In fact, considering the higher uncertainty involved in the users being imputed in this experiment, these results are very promising, since we are taking more risks by considering users with less information in their training set, hence being easier to fail in the imputations or to introduce noisy information.

5.3 Discussion

In this work, we have demonstrated through various experiments the beneficial impact of data imputation techniques on the performance of different recommender systems, especially in terms of ranking accuracy. Among the evaluated strategies, using the hybrid PGN algorithm to impute data stands out as the most promising option, offering substantial scope for improvement.

However, it is important to emphasize again that all the results presented so far have been obtained using the optimal hyperparameter configuration derived from the non-imputed version. In other words, once we determine the best parameter settings for each model using the non-imputed training file, we maintain these settings across all imputation strategies. This approach suggests that if we were to select hyperparameters using the imputed training file, we could potentially increase the performance of the algorithms even further. This improvement could be achieved by fine-tuning the parameters for each specific training set used.

To illustrate this, we present in Figure 14 a series of boxplots for the city of London (belonging to Foursquare), whereas those corresponding to Yelp and New York City and San Francisco (Gowalla) are shown in Appendix C (Figures 15, 16, and 17) for the sake of space. Each boxplot represents the performance obtained by each recommender considering all its hyperparameters (see Table 4) considering the original training set (red), the imputed training set using the PGN recommender with a 10% increment (blue), and the imputed training set with a 30% increment (green) for every training user. The median and average of each boxplot is represented with the horizontal black

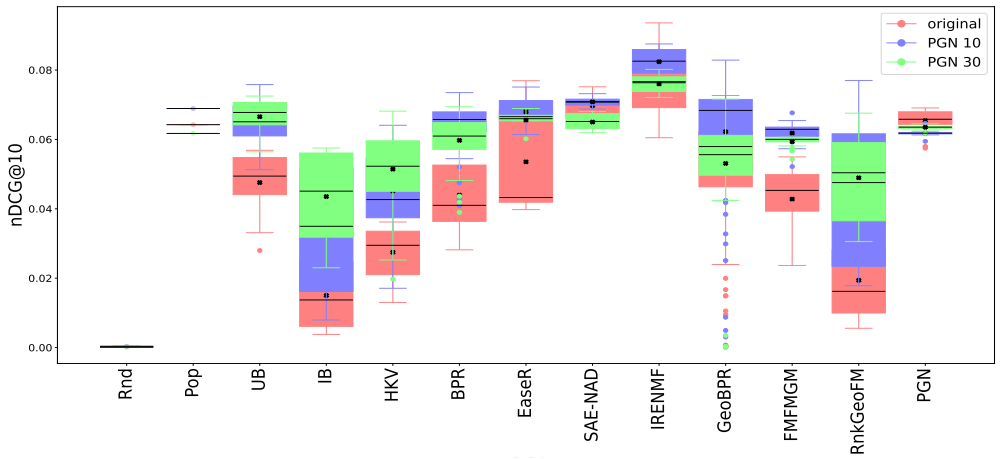

Fig. 14. Boxplots for the city of London (Foursquare) of the performance values obtained by different configurations of the recommendation algorithms considering all the hyperparameters. In red, the values obtained with no imputation. In blue, using the PGN as imputation algorithm, incrementing the number of check-ins by a 10% per user. In green, the same but incrementing the number of check-ins by a 30%.

lines and with the black "x", respectively. For example, the UB recommender has 20 different configurations (10 different neighbor values with 2 different similarities), hence, we compute a boxplot using the performance values obtained by the 20 versions of the UB recommender for every training set. Note that recommenders that do not have parameters, that is, Rnd and Pop do not produce a boxplot, as only 1 measurement for both the original training and the imputed versions would exist. Therefore, each boxplot encapsulates the performance of every parametrization for each recommender according to the nDCG@10 metric.

For Foursquare, in Figure 14, we observe that in most instances, the boxplots of the imputed versions outperform the original ones. This is particularly noticeable in the case of classical recommendation models that do not incorporate geographic influence, such as UB, IB, HKV, and BPR. When analyzing the POIs recommenders, there is one model that does not appear to benefit as significantly from the imputation, and three models that do not get their best result with imputed data (although imputation improves their results overall). The first one is the PGN model, which is, to some extent, expected, given that data imputation was conducted using the same algorithm, making it unlikely to benefit from its own imputation strategy. The other models are EASEr, SAE-NAD, and IRenMF, where most of their original performance values are outperformed when imputation is considered, except for their best value (top horizontal line). Furthermore, it is worth noting that we have conducted Wilcoxon statistical tests between the results of the non-imputed and imputed versions for all models. All of them, except IRenMF, achieved statistical significance (p-value < 0.05), underscoring the effectiveness of the imputation strategy.

Regarding the analysis for the other datasets, we refer to Appendix C for a detailed discussion. In summary, we observe that some models do not benefit from our imputation techniques as clearly as in Foursquare (as seen in London, Figure 14, where only PGN performance does not improve with imputation), while others do, so performance improvements seem to depend more on the type of approach than in Foursquare. Nonetheless, we would like to emphasize that our framework is highly configurable, and we find it very promising that, even with simple imputation strategies and straightforward parameter exploration, we can improve the performance of a wide range of

models. In the future work section, we outline how the quality of imputed data can be enhanced through more advanced strategies.

6 RELATED WORK

In the Point-of-Interest recommendation domain, as far as we know, the topic of data imputation remains relatively unexplored. While some studies have made progress in this area, it is essential to distinguish between approaches that enhance existing check-ins with additional POI information and those that infer new check-ins. Regarding the first type of proposals, we have the work of [9], where the authors introduced a neural network model to add Point-of-Interest information into social media posts. [31] proposed a similar approach based on RankSVM to address the scarcity of POI details in social media. On the other hand, some of the proposals aimed at generating new check-ins include the work of [33], where the authors proposed the POI-Augmentation Sequence-to-Sequence model (PA-Seq2S). This model, based on neural networks, is specifically designed to fill in potential gaps in user check-in sequences by predicting missing check-ins. SGRec from [32] is another interesting approach which uses the Sequence-to-Graph (Seq2Graph) augmentation technique, associating Point-of-Interest nodes with neighboring venues using semantic relationships to reduce data sparsity. Similarly, [71] uses a self-attention mechanism to complete trajectory sequences caused by gaps in user check-in data.

It should be noted that these works are designed for the next-Point-of-Interest recommendation problem, i.e., oriented to recommend the next Point-of-Interest to be visited by the user, and that they tested their proposals only against other sequential models; in comparison with our work here, which is designed for the original Point-of-Interest recommendation scenario. In fact, the generic nature of our framework makes it possible to incorporate such previous works, even though we leave this effort for the future, as it is out of the scope of our initial analysis of the problem. Additionally, it should be emphasized that the work presented herein is not tied to any particular recommendation algorithm or imputation strategy. We have tested methods with different characteristics and obtained promising results in most of the scenarios, evidencing that the POI recommendation problem could benefit from simple but domain-specific solutions.

7 CONCLUSIONS & FUTURE WORK

7.1 Summary

The Point-of-Interest recommendation domain has always been characterized by higher sparsity than other traditional recommendation areas, such as movies or books. The data available to train the models is generally limited, with many users making only a few check-ins, and few venues drawing a lot of the attention, evidencing a strong popularity bias. This results in a poor performance of models not only in terms of ranking accuracy but also in terms of novelty and diversity. In order to address this problem, in this work we have proposed a generic framework that allows us to model a data imputation process in the area of Point-of-Interest, by generating new check-ins for the recommenders while making use of a set of parameters grouped in two categories: parameters to configure how to choose the preferences to be imputed and parameters oriented to define the values to be imputed. These parameters allow to control how much data sparsity is reduced and which methods are used for this goal.

We have presented a set of experiments in six cities from the Foursquare global check-in dataset, two cities from Gowalla, and the Yelp dataset as a whole using two different models to impute data using collaborative filtering and geographic information approaches. As shown in the results, we have been able to improve the nDCG performance of most recommenders compared to their standard versions without imputation. Among the framework parameters, the most influential ones

towards the performance of the recommenders are the algorithms used for data imputation. Both PGN and IB strategies enable us to impute more useful data for the recommenders. Additionally, the imputation percentage plays a crucial role: generally, imputing 30% of the data yields better results than imputing only 10%, evidencing that data sparsity reduction is linked to better performance results. Furthermore, imputing data for all users tends to produce better outcomes than targeting only those with few check-ins.

We consider these results positive and very promising, especially since we did not perform hyper-parameter selection with the imputed check-in matrix C', as we maintained the same parameters as those obtained in the non-imputed training.

7.2 Limitations

Throughout this paper, we have emphasized the issue of data sparsity, and we have explored how imputation techniques can mitigate it, thereby enhancing the performance of recommendation algorithms. Nevertheless, despite the positive results achieved with our proposed methods, we believe this work can be further extended by integrating insights from other areas of recommender systems research.

One particularly promising area for such integration is the analysis of recommendation uncertainty, which is often discussed in terms of its opposite concepts: reliability and confidence [7, 61, 41]. In sparse situations (like POI recommendation) the algorithms infer the users preferences relying on small amount of data, making uncertain recommendations, and hence, measuring and/or estimating the confidence on recommender systems predictions is critical to detect which recommendations are more probable to be unsuccessful [15]. This confidence has been previously analyzed in CF techniques based on similarities and matrix factorization [6, 41] and also using approaches based on Bayesian Neural Networks and Monte-Carlo Markov Chains [14]. Therefore, for future work, we believe it is natural to integrate some of these techniques to determine whether the imputed data are sufficiently reliable to be included in the imputed matrix and be used later for generating the recommendations.

Imputation, while helpful for data completion, raises privacy concerns, particularly when dealing with sensitive user data in recommender systems and Point-of-Interest domain, such as the user's location. Imputing values can inadvertently reveal private information if the imputation method relies on correlations or patterns within the data. For instance, if a user's sensitive attribute is strongly correlated with other features, imputation could reconstruct that attribute, even if it was initially missing. Hence, this would require careful consideration of privacy-preserving imputation techniques and robust data anonymization practices to mitigate these risks.

Another major limitation of this work, but that can be extended to how recommender systems are evaluated nowadays [22], is the focus on short-term improvements in recommendation performance, as their long-term impact is not feasible to be assessed with offline experiments. Hence, further and more complete experiments should be designed to analyze the stability of imputed data and the performance changes in the long term.

7.3 Future work

Although the results shown here are promising, there is still room for improvement. As future work, we pretend to perform a more comprehensive analysis using more variables in the data imputation framework. Besides, we plan to make a more complete analysis of how these imputation techniques work for different user groups (national, international, local tourists, etc.), as it has been observed by the community that each user group exhibits different behaviors [52]. Similarly, after observing the positive results of using geographic information in our data imputation framework, we plan to include other types of information that we can find in LBSNs, such as temporal data and/or the

POI categories [53]. In fact, our results open up the possibility to analyze in detail how different types of POIs might be affected by the imputation framework, to consider whether some POIs are easier to impute than others based on the framework parameters.

At the same time, we consider exploring the effect of other imputation strategies on more complex and data intensive models such as neural networks and factorization machines, in order to detect whether these techniques can benefit from our framework and to what extent. Similarly, we intend to make use of such complex models to be used as imputation algorithms. However, especially when dealing with complex techniques, it is important to be aware of the privacy concerns mentioned in the previous section, and especially, to incorporate privacy-preserving methods to avoid leaking personal information.

Moreover, in the context of POI recommendation, imputing missing check-ins could benefit from cross-domain techniques [17] by incorporating data from other location-based social networks. For instance, if a user's check-in history is sparse in one dataset, information from another LBSN could help validate or refine imputed check-ins by identifying similar behavioral patterns across platforms. Moreover, cross-domain knowledge could serve as a filtering mechanism to ensure that imputed check-ins make sense; for example, if a POI appears frequently in one dataset but is rarely visited in another, this discrepancy might indicate potential noise in the imputed data. By integrating data from multiple sources, we could improve the accuracy of imputations and mitigate biases, ultimately leading to more robust POI recommendation models.

An alternative research line for the future would be to consider the causal structure of the data, so that the imputation mechanism would model the inherent biases and implicitly consider the underlying data structure, favoring domain generalization, as recently analyzed in [26]. In parallel, it is worth considering the quality of the imputed data. In principle, the goal of imputing data is to complete the user-venue matrix, meaning that the imputed check-ins should be useful to the subsequent recommendation algorithm. However, ensuring the quality of imputed data is a worthwhile avenue for the future. One potential approach we envision for this is the need of some kind of validation. In the domain presented in our work, this could be done by comparing the imputed check-ins with POI opening hours, user mobility patterns, and even weather conditions to assess if they would be valid, realistic check-ins performed by users; for example, a check-in at an outdoor location during extreme weather might indicate an unreliable imputation. Such additional validations present significant challenges but could enhance the reliability of imputed data and deserve further research efforts.

Finally, we plan to explore several optimization strategies to improve the scalability of the proposed framework for large-scale datasets. One promising direction is the use of more efficient sampling techniques during the imputation phase, prioritizing the imputation of interactions that are more likely to impact the training of the recommenders. Additionally, since exact computations are not always necessary, the use of approximate methods — such as approximate nearest neighbor [10] algorithms for similarity calculations — could significantly accelerate the imputation process without substantially sacrificing accuracy. Moreover, many components of the framework are well-suited for parallelization or distributed computing, which could further reduce execution time and make the approach more practical for real-world applications.

ACKNOWLEDGMENTS

This work was supported by grant PID2022-139131NB-I00 funded by MCIN/AEI/10.13039/ 501100011033 and by "ERDF A way of making Europe." The authors thank the reviewers for their thoughtful comments and suggestions.

REFERENCES

[1] Malika Acharya, Shilpi Yadav, and Krishna Kumar Mohbey. 2023. How can we create a recommender system for tourism? A location centric spatial binning-based methodology using social networks. *Int. J. Inf. Manag. Data Insights*, 3, 1, 100161. DOI: 10.1016/J.JJIMEI.2023.100161.

[2] Fatemah Alghamedy and Jun Zhang. 2019. Imputation strategies for cold-start users in nmf-based recommendation systems. In *Proceedings of the 3rd International Conference on Information System and Data Mining, ICISDM 2019, Houston, TX, USA, April 6-8, 2019*. ACM, 119–128. DOI: 10.1145/3325917.3325933.

[3] Abhijit Anand, Jurek Leonhardt, Jaspreet Singh, Koustav Rudra, and Avishek Anand. 2024. Data augmentation for sample efficient and robust document ranking. *ACM Trans. Inf. Syst.*, 42, 5, 119:1–119:29. DOI: 10.1145/3634911.

[4] Hong-Kyun Bae, Hyung-Ook Kim, Won-Yong Shin, and Sang-Wook Kim. 2021. "how to get consensus with neighbors?": rating standardization for accurate collaborative filtering. *Knowl. Based Syst.*, 234, 107549. DOI: 10.1016/J.KNOSYS.2021.107549.

[5] Alejandro Bellogín and Alan Said. 2021. Improving accountability in recommender systems research through reproducibility. *User Model. User Adapt. Interact.*, 31, 5, 941–977. DOI: 10.1007/S11257-021-09302-X.

[6] Cesare Bernardis, Maurizio Ferrari Dacrema, and Paolo Cremonesi. 2019. Estimating confidence of individual user predictions in item-based recommender systems. In *Proceedings of the 27th ACM Conference on User Modeling, Adaptation and Personalization, UMAP 2019, Larnaca, Cyprus, June 9-12, 2019*. George Angelos Papadopoulos, George Samaras, Stephan Weibelzahl, Dietmar Jannach, and Olga C. Santos, (Eds.) ACM, 149–156. DOI: 10.1145/3320435.3320453.

[7] Jesús Bobadilla, Abraham Gutiérrez, Fernando Ortega, and Bo Zhu. 2018. Reliability quality measures for recommender systems. *Inf. Sci.*, 442-443, 145–157. DOI: 10.1016/J.INS.2018.02.030.

[8] Pablo Castells, Neil Hurley, and Saúl Vargas. 2022. Novelty and diversity in recommender systems. In *Recommender Systems Handbook*. Francesco Ricci, Lior Rokach, and Bracha Shapira, (Eds.) Springer US, 603–646. DOI: 10.1007/978-1-0716-2197-4_16.

[9] Buru Chang, Yonggyu Park, Seongsoon Kim, and Jaewoo Kang. 2018. Deeppim: A deep neural point-of-interest imputation model. *Inf. Sci.*, 465, 61–71. DOI: 10.1016/J.INS.2018.06.065.

[10] Rihan Chen et al. 2022. Approximate nearest neighbor search under neural similarity metric for large-scale recommendation. In *Proceedings of the 31st ACM International Conference on Information & Knowledge Management, Atlanta, GA, USA, October 17-21, 2022*. Mohammad Al Hasan and Li Xiong, (Eds.) ACM, 3013–3022. DOI: 10.1145/3511808.3557098.

[11] Chen Cheng, Haiqin Yang, Irwin King, and Michael R. Lyu. 2012. Fused matrix factorization with geographical and social influence in location-based social networks. In *Proceedings of the Twenty-Sixth AAAI Conference on Artificial Intelligence, July 22-26, 2012, Toronto, Ontario, Canada*. Jörg Hoffmann and Bart Selman, (Eds.) AAAI Press. http://www.aaai.org/ocs/index.php/AAAI/AAAI12/paper/view/4748.

[12] Jin Yao Chin, Yile Chen, and Gao Cong. 2022. The datasets dilemma: how much do we really know about recommendation datasets? In *WSDM '22: The Fifteenth ACM International Conference on Web Search and Data Mining, Virtual Event / Tempe, AZ, USA, February 21 - 25, 2022*. K. Selcuk Candan, Huan Liu, Leman Akoglu, Xin Luna Dong, and Jiliang Tang, (Eds.) ACM, 141–149. DOI: 10.1145/3488560.3498519.

[13] Eunjoon Cho, Seth A. Myers, and Jure Leskovec. 2011. 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, San Diego, CA, USA, August 21-24, 2011*. Chid Apté, Joydeep Ghosh, and Padhraic Smyth, (Eds.) ACM, 1082–1090. DOI: 10.1145/2020408.2020579.

[14] Victor Coscrato and Derek Bridge. 2022. Recommendation uncertainty in implicit feedback recommender systems. In *Artificial Intelligence and Cognitive Science - 30th Irish Conference, AICS 2022, Munster, Ireland, December 8-9, 2022, Revised Selected Papers* (Communications in Computer and Information Science). Luca Longo and Ruairi O'Reilly, (Eds.) Vol. 1662. Springer, 279–291. DOI: 10.1007/978-3-031-26438-2_22.

[15] Victor Coscrato and Derek G. Bridge. 2023. Estimating and evaluating the uncertainty of rating predictions and top-n recommendations in recommender systems. *Trans. Recomm. Syst.*, 1, 2, 1–34. DOI: 10.1145/3584021.

[16] Maurizio Ferrari Dacrema, Simone Boglio, Paolo Cremonesi, and Dietmar Jannach. 2021. A troubling analysis of reproducibility and progress in recommender systems research. *ACM Trans. Inf. Syst.*, 39, 2, 20:1–20:49. DOI: 10.1145/3434185.

[17] Maurizio Ferrari Dacrema, Iván Cantador, Ignacio Fernández-Tobías, Shlomo Berkovsky, and Paolo Cremonesi. 2022. Design and evaluation of cross-domain recommender systems. In *Recommender Systems Handbook*. Francesco Ricci, Lior Rokach, and Bracha Shapira, (Eds.) Springer US, 485–516. DOI: 10.1007/978-1-0716-2197-4_13.

[18] Yizhou Dang, Enneng Yang, Guibing Guo, Linying Jiang, Xingwei Wang, Xiaoxiao Xu, Qinghui Sun, and Hong Liu. 2023. Uniform sequence better: time interval aware data augmentation for sequential recommendation. In *Thirty-Seventh AAAI Conference on Artificial Intelligence, AAAI 2023, Thirty-Fifth Conference on Innovative Applications of Artificial Intelligence, IAAI 2023, Thirteenth Symposium on Educational Advances in Artificial Intelligence, EAAI 2023,*

Washington, DC, USA, February 7-14, 2023. Brian Williams, Yiling Chen, and Jennifer Neville, (Eds.) AAAI Press, 4225–4232. DOI: 10.1609/AAAI.V37I4.25540.

[19] Mehdi Elahi, Francesco Ricci, and Neil Rubens. 2013. Active learning strategies for rating elicitation in collaborative filtering: A system-wide perspective. *ACM Trans. Intell. Syst. Technol.*, 5, 1, 13:1–13:33. DOI: 10.1145/2542182.2542195.

[20] Jean Mundahl Engels and Paula Diehr. 2003. Imputation of missing longitudinal data: a comparison of methods. *Journal of clinical epidemiology*, 56, 10, 968–976.

[21] Matthias Feurer, Aaron Klein, Katharina Eggensperger, Jost Tobias Springenberg, Manuel Blum, and Frank Hutter. 2019. Auto-sklearn: efficient and robust automated machine learning. In *Automated Machine Learning - Methods, Systems, Challenges*. The Springer Series on Challenges in Machine Learning. Frank Hutter, Lars Kotthoff, and Joaquin Vanschoren, (Eds.) Springer, 113–134. DOI: 10.1007/978-3-030-05318-5_6.

[22] Asela Gunawardana, Guy Shani, and Sivan Yogev. 2022. Evaluating recommender systems. In *Recommender Systems Handbook*. Francesco Ricci, Lior Rokach, and Bracha Shapira, (Eds.) Springer US, 547–601. DOI: 10.1007/978-1-0716-2197-4_15.

[23] Sajal Halder, Kwan Hui Lim, Jeffrey Chan, and Xiuzhen Zhang. 2022. POI recommendation with queuing time and user interest awareness. *Data Min. Knowl. Discov.*, 36, 6, 2379–2409. DOI: 10.1007/S10618-022-00865-W.

[24] Yifan Hu, Yehuda Koren, and Chris Volinsky. 2008. Collaborative filtering for implicit feedback datasets. In *Proceedings of the 8th IEEE International Conference on Data Mining (ICDM 2008), December 15-19, 2008, Pisa, Italy*. IEEE Computer Society, 263–272. DOI: 10.1109/ICDM.2008.22.

[25] Sunghyun Hwang and Dong-Kyu Chae. 2022. An uncertainty-aware imputation framework for alleviating the sparsity problem in collaborative filtering. In *Proceedings of the 31st ACM International Conference on Information & Knowledge Management, Atlanta, GA, USA, October 17-21, 2022*. Mohammad Al Hasan and Li Xiong, (Eds.) ACM, 802–811. DOI: 10.1145/3511808.3557236.

[26] Trent Kyono, Yao Zhang, Alexis Bellot, and Mihaela van der Schaar. 2021. MIRACLE: causally-aware imputation via learning missing data mechanisms. In *Advances in Neural Information Processing Systems 34: Annual Conference on Neural Information Processing Systems 2021, NeurIPS 2021, December 6-14, 2021, virtual*. Marc'Aurelio Ranzato, Alina Beygelzimer, Yann N. Dauphin, Percy Liang, and Jennifer Wortman Vaughan, (Eds.), 23806–23817. https://proceedings.neurips.cc/paper/2021/hash/c80bcf42c220b8f5c41f85344242f1b0-Abstract.html.

[27] Yeon-Chang Lee and Sang-Wook Kim. 2023. Uninteresting items: concept and its application to effective collaborative filtering in recommender systems. *SIGWEB Newsl.*, 2023, Autumn, 4:1–4:13. DOI: 10.1145/3631358.3631362.

[28] Youngnam Lee, Sang-Wook Kim, Sunju Park, and Xing Xie. 2018. How to impute missing ratings?: claims, solution, and its application to collaborative filtering. In *Proceedings of the 2018 World Wide Web Conference on World Wide Web, WWW 2018, Lyon, France, April 23-27, 2018*. Pierre-Antoine Champin, Fabien Gandon, Mounia Lalmas, and Panagiotis G. Ipeirotis, (Eds.) ACM, 783–792. DOI: 10.1145/3178876.3186159.

[29] Peibo Li, Maarten de Rijke, Hao Xue, Shuang Ao, Yang Song, and Flora D. Salim. 2024. Large language models for next point-of-interest recommendation. In *Proceedings of the 47th International ACM SIGIR Conference on Research and Development in Information Retrieval, SIGIR 2024, Washington DC, USA, July 14-18, 2024*. Grace Hui Yang, Hongning Wang, Sam Han, Claudia Hauff, Guido Zuccon, and Yi Zhang, (Eds.) ACM, 1463–1472. DOI: 10.1145/3626772.3657840.

[30] Xutao Li, Gao Cong, Xiaoli Li, Tuan-Anh Nguyen Pham, and Shonali Krishnaswamy. 2015. Rank-geofm: A ranking based geographical factorization method for point of interest recommendation. In *Proceedings of the 38th International ACM SIGIR Conference on Research and Development in Information Retrieval, Santiago, Chile, August 9-13, 2015*. Ricardo A. Baeza-Yates, Mounia Lalmas, Alistair Moffat, and Berthier A. Ribeiro-Neto, (Eds.) ACM, 433–442. DOI: 10.1145/2766462.2767722.

[31] Xutao Li, Tuan-Anh Nguyen Pham, Gao Cong, Quan Yuan, Xiaoli Li, and Shonali Krishnaswamy. 2015. Where you instagram?: associating your instagram photos with points of interest. In *Proceedings of the 24th ACM International Conference on Information and Knowledge Management, CIKM 2015, Melbourne, VIC, Australia, October 19 - 23, 2015*. James Bailey, Alistair Moffat, Charu C. Aggarwal, Maarten de Rijke, Ravi Kumar, Vanessa Murdock, Timos K. Sellis, and Jeffrey Xu Yu, (Eds.) ACM, 1231–1240. DOI: 10.1145/2806416.2806463.

[32] Yang Li, Tong Chen, Yadan Luo, Hongzhi Yin, and Zi Huang. 2021. Discovering collaborative signals for next POI recommendation with iterative seq2graph augmentation. In *Proceedings of the Thirtieth International Joint Conference on Artificial Intelligence, IJCAI 2021, Virtual Event / Montreal, Canada, 19-27 August 2021*. Zhi-Hua Zhou, (Ed.) ijcai.org, 1491–1497. DOI: 10.24963/IJCAI.2021/206.

[33] Yang Li, Yadan Luo, Zheng Zhang, Shazia W. Sadiq, and Peng Cui. 2019. Context-aware attention-based data augmentation for POI recommendation. In *35th IEEE International Conference on Data Engineering Workshops, ICDE Workshops 2019, Macao, China, April 8-12, 2019*. IEEE, 177–184. DOI: 10.1109/ICDEW.2019.00-14.

[34] Yang Liu and An-bo Wu. 2021. POI recommendation method using deep learning in location-based social networks. *Wirel. Commun. Mob. Comput.*, 2021, 9120864:1–9120864:11. DOI: 10.1155/2021/9120864.

[35] Yong Liu, Wei Wei, Aixin Sun, and Chunyan Miao. 2014. Exploiting geographical neighborhood characteristics for location recommendation. In *Proceedings of the 23rd ACM International Conference on Conference on Information and Knowledge Management, CIKM 2014, Shanghai, China, November 3-7, 2014*. Jianzhong Li, Xiaoyang Sean Wang, Minos N. Garofalakis, Ian Soboroff, Torsten Suel, and Min Wang, (Eds.) ACM, 739–748. DOI: 10.1145/2661829.2662002.

[36] Jing Long, Tong Chen, Quoc Viet Hung Nguyen, and Hongzhi Yin. 2023. Decentralized collaborative learning framework for next POI recommendation. *ACM Trans. Inf. Syst.*, 41, 3, 66:1–66:25. DOI: 10.1145/3555374.

[37] Chen Ma, Yingxue Zhang, Qinglong Wang, and Xue Liu. 2018. Point-of-interest recommendation: exploiting self-attentive autoencoders with neighbor-aware influence. In *Proceedings of the 27th ACM International Conference on Information and Knowledge Management, CIKM 2018, Torino, Italy, October 22-26, 2018*. Alfredo Cuzzocrea et al., (Eds.) ACM, 697–706. DOI: 10.1145/3269206.3271733.

[38] Kiran Maharana, Surajit Mondal, and Bhushankumar Nemade. 2022. A review: data pre-processing and data augmentation techniques. *Global Transitions Proceedings*, 3, 1, 91–99.

[39] Alex Martinez, Mihnea Tufis, and Ludovico Boratto. 2024. Unmasking privacy: A reproduction and evaluation study of obfuscation-based perturbation techniques for collaborative filtering. In *Proceedings of the 47th International ACM SIGIR Conference on Research and Development in Information Retrieval, SIGIR 2024, Washington DC, USA, July 14-18, 2024*. Grace Hui Yang, Hongning Wang, Sam Han, Claudia Hauff, Guido Zuccon, and Yi Zhang, (Eds.) ACM, 1753–1762. DOI: 10.1145/3626772.3657858.

[40] David Massimo and Francesco Ricci. 2022. Building effective recommender systems for tourists. *AI Mag.*, 43, 2, 209–224. DOI: 10.1002/AAAI.12057.

[41] Rus M. Mesas and Alejandro Bellogín. 2020. Exploiting recommendation confidence in decision-aware recommender systems. *J. Intell. Inf. Syst.*, 54, 1, 45–78. DOI: 10.1007/S10844-018-0526-3.

[42] Athanasios N. Nikolakopoulos, Xia Ning, Christian Desrosiers, and George Karypis. 2022. Trust your neighbors: A comprehensive survey of neighborhood-based methods for recommender systems. In *Recommender Systems Handbook*. Francesco Ricci, Lior Rokach, and Bracha Shapira, (Eds.) Springer US, 39–89. DOI: 10.1007/978-1-0716-2197-4_2.

[43] Massimo Quadrana, Paolo Cremonesi, and Dietmar Jannach. 2018. Sequence-aware recommender systems. *ACM Comput. Surv.*, 51, 4, 66:1–66:36. DOI: 10.1145/3190616.

[44] Hossein A. Rahmani, Mohammad Aliannejadi, Mitra Baratchi, and Fabio Crestani. 2022. A systematic analysis on the impact of contextual information on point-of-interest recommendation. *ACM Trans. Inf. Syst.*, 40, 4, 88:1–88:35. DOI: 10.1145/3508478.

[45] Manizheh Ranjbar, Parham Moradi, Mostafa Azami, and Mahdi Jalili. 2015. An imputation-based matrix factorization method for improving accuracy of collaborative filtering systems. *Eng. Appl. Artif. Intell.*, 46, 58–66. DOI: 10.1016/J.ENGAPPAI.2015.08.010.

[46] Yongli Ren, Gang Li, Jun Zhang, and Wanlei Zhou. 2013. Adam: adaptive-maximum imputation for neighborhood-based collaborative filtering. In *Advances in Social Networks Analysis and Mining 2013, ASONAM '13, Niagara, ON, Canada - August 25 - 29, 2013*. Jon G. Rokne and Christos Faloutsos, (Eds.) ACM, 628–635. DOI: 10.1145/2492517.2492565.

[47] Yongli Ren, Gang Li, Jun Zhang, and Wanlei Zhou. 2012. The efficient imputation method for neighborhood-based collaborative filtering. In *21st ACM International Conference on Information and Knowledge Management, CIKM'12, Maui, HI, USA, October 29 - November 02, 2012*. Xue-wen Chen, Guy Lebanon, Haixun Wang, and Mohammed J. Zaki, (Eds.) ACM, 684–693. DOI: 10.1145/2396761.2396849.

[48] Yongli Ren, Gang Li, Jun Zhang, and Wanlei Zhou. 2014. The maximum imputation framework for neighborhood-based collaborative filtering. *Soc. Netw. Anal. Min.*, 4, 1, 207. DOI: 10.1007/S13278-014-0207-3.

[49] Steffen Rendle, Christoph Freudenthaler, Zeno Gantner, and Lars Schmidt-Thieme. 2009. BPR: bayesian personalized ranking from implicit feedback. In *UAI 2009, Proceedings of the Twenty-Fifth Conference on Uncertainty in Artificial Intelligence, Montreal, QC, Canada, June 18-21, 2009*. Jeff A. Bilmes and Andrew Y. Ng, (Eds.) AUAI Press, 452–461.

[50] Francesco Ricci, Lior Rokach, and Bracha Shapira, (Eds.) 2022. *Recommender Systems Handbook*. Springer US. ISBN: 978-1-0716-2196-7. DOI: 10.1007/978-1-0716-2197-4.

[51] Alan Said and Alejandro Bellogín. 2014. Comparative recommender system evaluation: benchmarking recommendation frameworks. In *Eighth ACM Conference on Recommender Systems, RecSys '14, Foster City, Silicon Valley, CA, USA - October 06 - 10, 2014*. Alfred Kobsa, Michelle X. Zhou, Martin Ester, and Yehuda Koren, (Eds.) ACM, 129–136. DOI: 10.1145/2645710.2645746.

[52] Pablo Sánchez and Alejandro Bellogín. 2021. On the effects of aggregation strategies for different groups of users in venue recommendation. *Inf. Process. Manag.*, 58, 5, 102609. DOI: 10.1016/j.ipm.2021.102609.

[53] Pablo Sánchez and Alejandro Bellogín. 2022. Point-of-interest recommender systems based on location-based social networks: A survey from an experimental perspective. *ACM Comput. Surv.*, 54, 11s, 223:1–223:37. DOI: 10.1145/3510409.

[54] Pablo Sánchez, Alejandro Bellogín, and Ludovico Boratto. 2023. Bias characterization, assessment, and mitigation in location-based recommender systems. *Data Min. Knowl. Discov.*, 37, 5, 1885–1929. DOI: 10.1007/S10618-022-00913-5.

[55] Shubhra Kanti Karmaker Santu, Md. Mahadi Hassan, Micah J. Smith, Lei Xu, Chengxiang Zhai, and Kalyan Veeramachaneni. 2022. Automl to date and beyond: challenges and opportunities. *ACM Comput. Surv.*, 54, 8, 175:1–175:36. DOI: 10.1145/3470918.

[56] Anna Sepliarskaia, Julia Kiseleva, Filip Radlinski, and Maarten de Rijke. 2018. Preference elicitation as an optimization problem. In *Proceedings of the 12th ACM Conference on Recommender Systems, RecSys 2018, Vancouver, BC, Canada, October 2-7, 2018*. Sole Pera, Michael D. Ekstrand, Xavier Amatriain, and John O'Donovan, (Eds.) ACM, 172–180. DOI: 10.1145/3240323.3240352.

[57] Yali Si, Fuzhi Zhang, and Wenyuan Liu. 2019. An adaptive point-of-interest recommendation method for location-based social networks based on user activity and spatial features. *Knowl.-Based Syst.*, 163, 267–282. DOI: 10.1016/j.knosys.2018.08.031.

[58] Harald Steck. 2019. Embarrassingly shallow autoencoders for sparse data. In *The World Wide Web Conference, WWW 2019, San Francisco, CA, USA, May 13-17, 2019*. Ling Liu, Ryen W. White, Amin Mantrach, Fabrizio Silvestri, Julian J. McAuley, Ricardo Baeza-Yates, and Leila Zia, (Eds.) ACM, 3251–3257. DOI: 10.1145/3308558.3313710.

[59] Xiaoyuan Su, Taghi M. Khoshgoftaar, Xingquan Zhu, and Russell Greiner. 2008. Imputation-boosted collaborative filtering using machine learning classifiers. In *Proceedings of the 2008 ACM Symposium on Applied Computing (SAC), Fortaleza, Ceara, Brazil, March 16-20, 2008*. Roger L. Wainwright and Hisham Haddad, (Eds.) ACM, 949–950. DOI: 10.1145/1363686.1363903.

[60] Saul Vargas and Pablo Castells. 2011. Rank and relevance in novelty and diversity metrics for recommender systems. In *Proceedings of the 2011 ACM Conference on Recommender Systems, RecSys 2011, Chicago, IL, USA, October 23-27, 2011*. Bamshad Mobasher, Robin D. Burke, Dietmar Jannach, and Gediminas Adomavicius, (Eds.) ACM, 109–116. DOI: 10.1145/2043932.2043955.

[61] Chao Wang, Qi Liu, Run-ze Wu, Enhong Chen, Chuanren Liu, Xunpeng Huang, and Zhenya Huang. 2018. Confidence-aware matrix factorization for recommender systems. In *Proceedings of the Thirty-Second AAAI Conference on Artificial Intelligence, (AAAI-18), the 30th innovative Applications of Artificial Intelligence (IAAI-18), and the 8th AAAI Symposium on Educational Advances in Artificial Intelligence (EAAI-18), New Orleans, Louisiana, USA, February 2-7, 2018*. Sheila A. McIlraith and Kilian Q. Weinberger, (Eds.) AAAI Press, 434–442. DOI: 10.1609/AAAI.V32I1.11251.

[62] Haiyang Wang, Yan Chu, Hui Ning, Zhengkui Wang, and Wen Shan. 2023. User feedback-based counterfactual data augmentation for sequential recommendation. In *Knowledge Science, Engineering and Management - 16th International Conference, KSEM 2023, Guangzhou, China, August 16-18, 2023, Proceedings, Part III* (Lecture Notes in Computer Science). Zhi Jin, Yuncheng Jiang, Robert Andrei Buchmann, Yaxin Bi, Ana-Maria Ghiran, and Wenjun Ma, (Eds.) Vol. 14119. Springer, 370–382. DOI: 10.1007/978-3-031-40289-0_30.

[63] Wei Wang, Junyang Chen, Jinzhong Wang, Junxin Chen, Jinquan Liu, and Zhiguo Gong. 2020. Trust-enhanced collaborative filtering for personalized point of interests recommendation. *IEEE Trans. Ind. Informatics*, 16, 9, 6124–6132. DOI: 10.1109/TII.2019.2958696.

[64] Weiwei Xia, Liang He, Junzhong Gu, and Keqin He. 2009. Effective collaborative filtering approaches based on missing data imputation. In *International Conference on Networked Computing and Advanced Information Management, NCM 2009, Fifth International Joint Conference on INC, IMS and IDC: INC 2009: International Conference on Networked Computing, IMS 2009: International Conference on Advanced Information Management and Service, IDC 2009: International Conference on Digital Content, Multimedia Technology and its Applications, Seoul, Korea, August 25-27, 2009*. Jinhwa Kim, Dursun Delen, Jinsoo Park, Franz Ko, Chen Rui, Jong Hyung Lee, Jian Wang, and Gang Kou, (Eds.) IEEE Computer Society, 534–537. DOI: 10.1109/NCM.2009.128.

[65] Carl Yang, Lanxiao Bai, Chao Zhang, Quan Yuan, and Jiawei Han. 2017. Bridging collaborative filtering and semi-supervised learning: A neural approach for POI recommendation. In *Proceedings of the 23rd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, Halifax, NS, Canada, August 13 - 17, 2017*. ACM, 1245–1254. DOI: 10.1145/3097983.3098094.

[66] Dingqi Yang, Daqing Zhang, and Bingqing Qu. 2016. Participatory cultural mapping based on collective behavior data in location-based social networks. *ACM TIST*, 7, 3, 30:1–30:23. DOI: 10.1145/2814575.

[67] Fajie Yuan, Joemon M. Jose, Guibing Guo, Long Chen, Haitao Yu, and Rami Suleiman Alkhawaldeh. 2016. Joint geo-spatial preference and pairwise ranking for point-of-interest recommendation. In *28th IEEE International Conference on Tools with Artificial Intelligence, ICTAI 2016, San Jose, CA, USA, November 6-8, 2016*. IEEE Computer Society, 46–53. DOI: 10.1109/ICTAI.2016.0018.

[68] Pengpeng Zhao, Anjing Luo, Yanchi Liu, Jiajie Xu, Zhixu Li, Fuzhen Zhuang, Victor S. Sheng, and Xiaofang Zhou. 2022. Where to go next: A spatio-temporal gated network for next POI recommendation. *IEEE Trans. Knowl. Data Eng.*, 34, 5, 2512–2524. DOI: 10.1109/TKDE.2020.3007194.

[69] Ruiqi Zheng, Liang Qu, Bin Cui, Yuhui Shi, and Hongzhi Yin. 2023. Automl for deep recommender systems: A survey. *ACM Trans. Inf. Syst.*, 41, 4, 101:1–101:38. DOI: 10.1145/3579355.

[70] Xu Zhou, Zhuoran Wang, Xuejie Liu, Yanheng Liu, and Geng Sun. 2024. An improved context-aware weighted matrix factorization algorithm for point of interest recommendation in LBSN. *Inf. Syst.*, 122, 102366. DOI: 10.1016/J.IS.2024.102366.

[71] Zhuang Zhuang, Tianxin Wei, Lingbo Liu, Heng Qi, Yanming Shen, and Baocai Yin. 2024. TAU: trajectory data augmentation with uncertainty for next POI recommendation. In *Thirty-Eighth AAAI Conference on Artificial Intelligence, AAAI 2024, Thirty-Sixth Conference on Innovative Applications of Artificial Intelligence, IAAI 2024, Fourteenth Symposium on Educational Advances in Artificial Intelligence, EAAI 2014, February 20-27, 2024, Vancouver, Canada*. Michael J. Wooldridge, Jennifer G. Dy, and Sriraam Natarajan, (Eds.) AAAI Press, 22565–22573. DOI: 10.1609/AAAI.V38I20.30265.

A PARAMETERS OF RECOMMENDATION ALGORITHMS

Table 4 includes the parameters used in the recommenders.

Table 4. Parameters of evaluated recommenders; the values that are not between the symbols {} are considered fixed and not tuned. Third column indicates the abbreviation used in tables and figures.

Recommender	**Abbr.**	**Parameters**
Popularity	Pop	None
Random	Rnd	None
User-based nearest neighbor	UB	k = {10, 20, 30, 40, 50, 60, 70, 80, 90, 100}, sim = {SetJ, VecC }
Item-based nearest neighbor	IB	k = {10, 20, 30, 40, 50, 60, 70, 80, 90, 100}, sim = {SetJ, VecC }
MF using Alternate Least Squares	HKV	k = {10, 50, 100}, α = {0.1, 1, 10}, λ = {0.1, 1, 10}
MF with Bayesian Personalized Ranking	BPR	k = {10, 50, 100}, $\lambda_u = \lambda_i$ = {0.001, 0.0025, 0.005, 0.01, 0.1}, λ_0 = {0, 0.5, 1}, $\lambda_j = \lambda_u/10$, iter = 50
Embarrassingly Shallow Autoencoders for Sparse Data	EASEr	λ = {0.5, 200, 500} Implicit = {True, False}
Self-attentive encoder and neighbor-aware decoder	SAE-NAD	epoch=20, batch=256, α = 2, $\epsilon = 1e-5$, learning_rate=1e-3, decay=1e-3, num_attentions={10, 20, 40}, dropout=0.5, γ = {30, 60, 90}
Instance-Region Neighborhood Matrix Factorization	IRenMF	Factors = {50, 100}, geo-α = {0.4, 0.6}, λ_3 = {0.1, 1}, clusters = {5, 50}
Geographical BPR	GeoBPR	k = {10, 50, 100}, $\lambda_u = \lambda_i$ = {0.001, 0.0025, 0.005, 0.01, 0.1}, λ_0 = {0, 0.5, 1}, $\lambda_j = \lambda_u/10$, iter = 50, max_dist = {1, 4}
Fused matrix factorization with Multi-center Gaussian Model	FMFMGM	α = {0.2, 0.4}, θ = {0.02, 0.2}, distance = 15, item = 30, factors = {50, 100}, α_2 = {20, 40}, β = 0.2, sigmoid = false, learning_rate = 0.0001
Ranking Geographical Factorization	RnkGeoFM	k = {10, 50, 100}, α = {0.1, 0.2}, n = {10, 50, 100, 200}, C = 1, ϵ = 0.3, iters = {50, 120, 200}, boldDriver = true, learn_rate= 0.001
Hybrid: Pop + UB + AvgDis	PGN	k = {10, 20, 30, 40, 50, 60, 70, 80, 90, 100}, sim = {SetJ, VecC }

Table 5. Number of times each imputation configuration achieves a performance equal to or greater than that of the "original" configuration (no imputation) in terms of nDCG@10 across all the cities of the different datasets, ignoring the Rnd, recommenders. This table represents a summary of Figures 4, 5, and 6.

Configuration	LON	NYC	SAN	MEX	MOS	TOK	Yelp	NY-G	SF-G
AvgDis 10	2	5	6	7	3	4	1	5	3
AvgDis 30	4	2	8	7	7	1	1	4	2
IB 10	3	2	5	6	9	4	1	1	0
IB 30	2	1	9	8	9	2	1	1	0
PGN 10	8	5	8	8	5	5	2	4	3
PGN 30	5	3	9	9	9	2	1	4	2
Rnd 10	3	5	3	6	6	2	3	2	3
Rnd 30	2	2	9	7	7	1	1	3	3

Table 6. Number of times each imputation configuration achieves a performance equal to or greater than that of the "original" configuration (no imputation) in terms of EPC @10 across all the cities, ignoring the Rnd recommender. This table represents a summary of Figures 7, 8, 9, and 10 for the EPC metric.

Configuration	LON	NYC	SAN	MEX	MOS	TOK	Yelp	NY-G	SF-G
AvgDis 10	11	11	10	6	10	10	12	9	10
AvgDis 30	11	11	10	9	12	11	12	12	12
IB 10	7	8	6	3	5	3	10	9	11
IB 30	8	8	7	6	6	3	10	10	9
PGN 10	2	3	2	1	2	2	2	3	3
PGN 30	2	4	2	2	1	1	2	5	2
Rnd 10	8	10	10	7	8	8	10	9	11
Rnd 30	10	10	11	6	9	10	11	10	11

B ADDITIONAL PERFORMANCE RESULTS OF IMPUTATION CONFIGURATIONS

Tables 5, 6, and 7 present the number of models whose imputed version improves the performance of the non-imputed version in all datasets in terms of nDCG, EPC, and Gini, respectively. Table 8 also shows the number of models where the imputation methodology improves the performance of the non-imputed version in terms of nDCG but in the cold-start scenario presented in Section 5.2.

C ADDITIONAL ANALYSIS OF HYPERPARAMETER TUNING APPLIED TO IMPUTATION STRATEGIES

This section presents additional results to those discussed in Section 5.3, corresponding to Yelp (Figure 15) and New York City (Figure 16) and San Francisco (Figure 17) from Gowalla. In both Gowalla cities, models such as RnkGeoFM, FMFMGM, HKV, and IB show a performance increase. For Yelp, the behavior is more similar to the results reported for Foursquare, although the performance gains are again slightly smaller. Additionally, statistical significance tests (Wilcoxon) reveal meaningful differences between models in the rest of the cities. For SF-G, all models achieved statistical significance (p-value < 0.05), while for NY-G, the results reported by BPR, UB, and GeoBPR are not statistically different from their non-imputed counterparts. Finally, for Yelp, the results obtained by HKV, IRenMF, SAE-NAD, and EASEr with data imputation are not statistically different from their non-imputed versions.

Table 7. Number of times each imputation configuration achieves a performance equal to or greater than that of the "original" configuration (no imputation) in terms of Gini@10 across all the cities, ignoring the Rnd recommender. This table represents a summary of Figures 7, 8, 9, and 10 for the Gini metric.

Configuration	LON	NYC	SAN	MEX	MOS	TOK	Yelp	NY-G	SF-G
AvgDis 10	12	11	9	7	8	9	12	10	10
AvgDis 30	11	11	11	11	11	11	12	12	11
IB 10	8	10	4	6	7	3	3	9	6
IB 30	9	10	5	5	7	5	3	8	6
PGN 10	3	3	2	4	2	2	1	4	1
PGN 30	2	3	0	0	1	1	1	1	0
Rnd 10	8	9	7	8	5	6	10	9	9
Rnd 30	11	10	10	8	7	6	11	9	11

Table 8. Number of times each imputation configuration achieves a performance equal to or greater than that of the "original" configuration (no imputation) in terms of nDCG@10 across the different cities for cold-start users, ignoring the Rnd recommender. This table represents a summary of Figures 11, 12, and 13.

Configuration	LON	NYC	SAN	MEX	MOS	TOK	Yelp	NY-G	SF-G
AvgDis 10	3	3	3	9	3	2	3	3	1
AvgDis 30	3	2	9	7	7	2	2	3	1
IB 10	4	3	3	9	9	4	5	1	1
IB 30	6	2	9	9	9	2	3	1	1
PGN 10	5	5	3	4	5	4	7	2	2
PGN 30	5	3	9	8	9	2	6	2	3
Rnd 10	6	3	2	9	6	5	0	4	4
Rnd 30	3	2	9	7	7	1	0	2	1

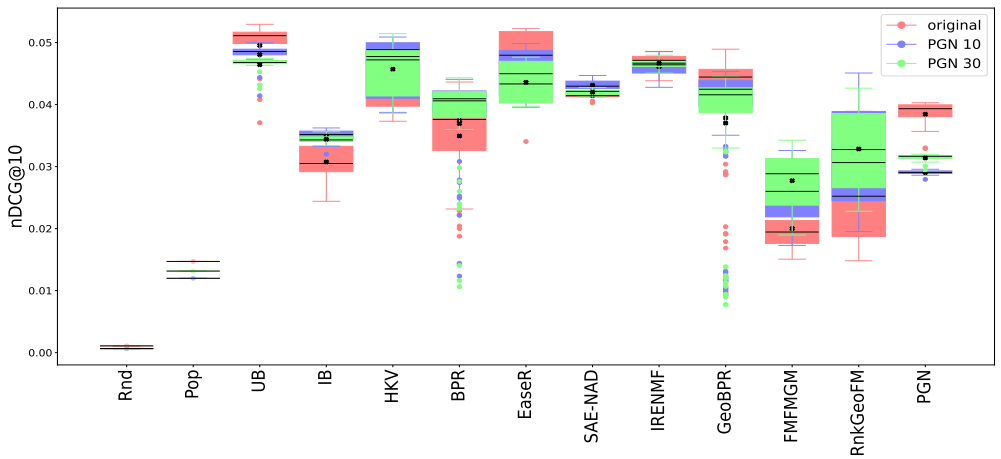

Fig. 15. Boxplots for Yelp of the performance values obtained by different configurations of the recommendation algorithms considering all the hyperparameters. Same configuration as in Figure 14.

Nevertheless, these differences in performance across datasets can be explained by the inherent characteristics of these datasets, as evidenced in the statistics reported in Table 1. For the case of

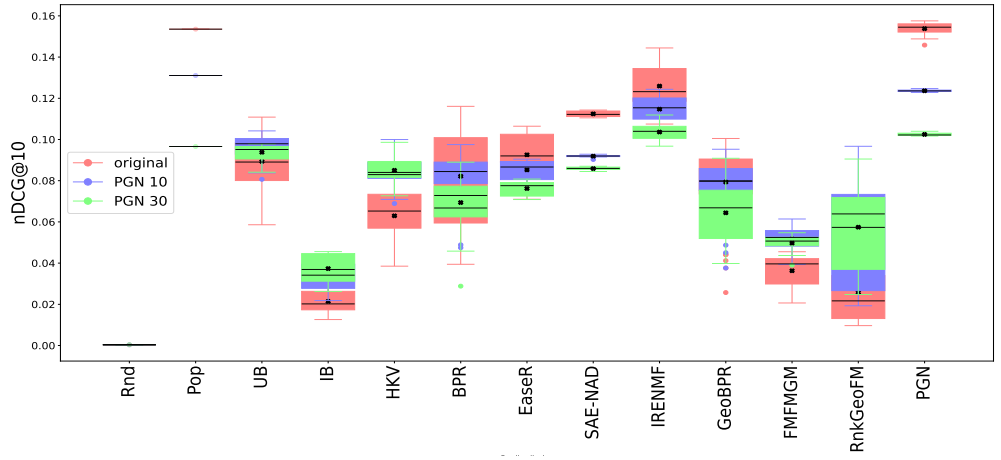

Fig. 16. Boxplots for New York City (Gowalla) of the performance values obtained by different configurations of the recommendation algorithms considering all the hyperparameters. Same configuration as in Figure 14.

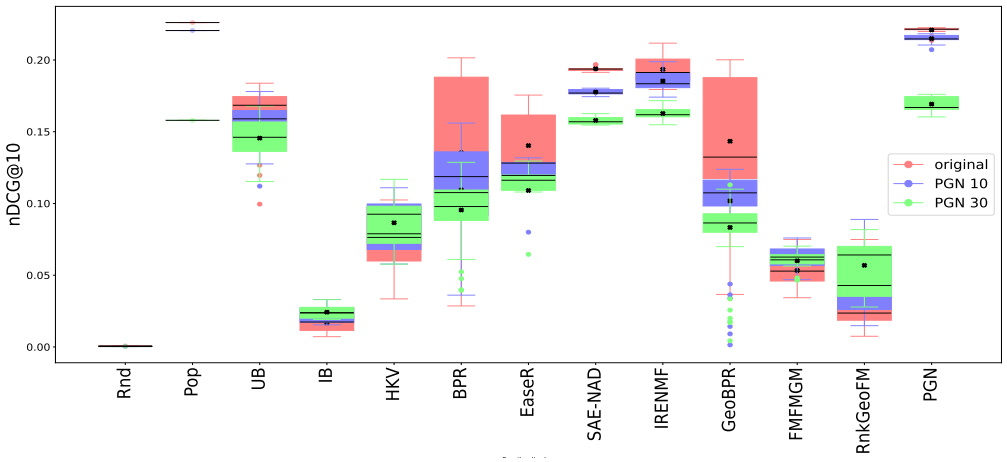

Fig. 17. Boxplots for San Francisco (Gowalla) of the performance values obtained by different configurations of the recommendation algorithms considering all the hyperparameters. Same configuration as in Figure 14.

Foursquare, the data is significantly sparser than in Gowalla and Yelp, which makes the imputed data more helpful. On the other hand, in our current imputation strategy, we only impute binary interactions (i.e., 1s), as defined by parameter $\mathcal{V}$ in our framework. As a result, if there are "real" check-ins with higher aggregated values (e.g., frequency), the imputed data may contribute less useful information, or even degrade model performance in some cases.