

A novel approach for venue recommendation using cross-domain techniques

Pablo Sánchez

Universidad Autónoma de Madrid
Madrid, Spain
pablo.sanchezp@uam.es

Alejandro Bellogín

Universidad Autónoma de Madrid
Madrid, Spain
alejandro.bellogin@uam.es

ABSTRACT

Finding the next venue to be visited by a user in a specific city is an interesting, but challenging, problem. Different techniques have been proposed, combining collaborative, content, social, and geographical signals; however it is not trivial to decide which technique works best, since this may depend on the data density or the amount of activity logged for each user or item. At the same time, cross-domain strategies have been exploited in the recommender systems literature when dealing with (very) sparse situations, such as those inherently arising when recommendations are produced based on information from a single city.

In this paper, we address the problem of venue recommendation from a novel perspective: applying cross-domain recommendation techniques considering each city as a different domain. We perform an experimental comparison of several recommendation techniques in a temporal split under two conditions: single-domain (only information from the target city is considered) and cross-domain (information from many other cities is incorporated into the recommendation algorithm). For the latter, we have explored two strategies to transfer knowledge from one domain to another: testing the target city and training a model with information of the k cities with more ratings or only using the k closest cities.

Our results show that, in general, applying cross-domain by proximity increases the performance of the majority of the recommenders in terms of relevance. This is the first work, to the best of our knowledge, where so many domains (eight) are combined in the tourism context where a temporal split is used, and thus we expect these results could provide readers with an overall picture of what can be achieved in a real-world environment.

1 INTRODUCTION

The great development of location-based social networks (LBSNs) in recent years has encouraged the research into the problem of Point-of-Interest (POI) or venue recommendation, i.e., suggesting new places to visit by analyzing the users' tastes, needs, and movement patterns. Foursquare, Gowalla, or GeoLife are examples of these kind of social networks, where users record check-ins they make to certain venues (restaurants, cinemas, hotels, etc.) and share

their experiences with other users in the system [28, 30]. This information, if processed and exploited correctly, can be then used to suggest new venues to visit when using a recommendation engine.

Since research on Recommender Systems (RS) has increased in many directions in the last years, it is important to consider some specific details of POI recommendation that differ from the more traditional recommendation problem [14, 17, 25]. These include, but are not limited, to:

- **Sparsity:** in the RS domain, the user-item matrix is normally very sparse. However, in venue recommendation, this effect is even more severe. For example, for the Movielens20M and Netflix datasets their densities are 0.539% and 1.177%, respectively. For the dataset we shall be using in our experiments (from Foursquare), the density is 0.0034%.
- **Implicit information:** check-in data is one type of implicit feedback, where only positive values indicating that a user has visited a venue are recorded. Nonetheless, since users may check-in in the same place several times, researchers often build frequency matrices to model these repetitions. This differs from traditional recommendation, where it is normally assumed that users rate each item once [20].
- **External influences:** venue recommendation is highly affected by social (user friends), temporal, and geographical influence. The latter is possibly the most important effect to take into account in POI recommendation to improve the recommendation performance as it is usually assumed that users prefer to visit venues that are close to each other (as the first law of geography states “*Everything is related to everything else, but near things are more related than distant things*” [10]). That is the reason why researchers have proposed many algorithms including some component of geographical influence [16, 18, 28].

At the same time, different techniques can be explored to obtain or access more information in order to help recommenders with their suggestions. Transfer (or cross-domain) learning is one of these valuable techniques that allow to make use of external or additional information, and integrate it with current knowledge. In the context of RS, cross-domain recommendation is a recent and active research topic, where tourism has been acknowledged as a potential target domain [33]. However, not many experimental comparisons have been performed using cross-domain techniques in the tourism field with a realistic evaluation, especially exploiting geographical or social influence.

Thus, based on the fact that there has not been an exhaustive analysis of the effect of different cross-domain strategies in venue recommendation, in this paper we analyze the effect of producing recommendations by using only the venues of each separate city

Permission to make digital or hard copies of part or all of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for third-party components of this work must be honored. For all other uses, contact the owner/author(s).
RecSysKTL'18, October 2018, Vancouver, Canada
© 2018 Copyright held by the owner/author(s).

(single-domain) and contrast it against the combination of information extracted from other cities (cross-domain). Therefore, in this paper we address the following research questions:

RQ1: Are state-of-the-art recommendation algorithms able to exploit cross-domain information in the context of venue recommendation? We empirically compare a set of recommenders under two conditions: single-domain (only information from the target city is considered) and cross-domain (information from many other cities is incorporated into the recommendation algorithm). We further explore different ways to select other cities to learn from in the cross-domain condition – more specifically, according to their popularity and the distance to the target city.

RQ2: Which is the best cross-domain strategy in terms of relevance? We focus on nDCG as ranking metric and see which cross-domain strategy has a larger impact in performance with respect to the single-domain approach.

Our work provides a thorough comparison of two cross-domain strategies for venue recommendation under a realistic evaluation by means of a time-aware methodology. We compare the results obtained by the cross-domain strategies against the single-domain scenario in terms of ranking accuracy using a real dataset with more than 30M interactions. Our results indicate that cross-domain strategies can improve the performance of some recommendation approaches, whereas for other recommendation algorithms – such as those based on geographical information – single-domain scenarios seem to be more appropriate.

2 BACKGROUND

In the RS community, a large number of algorithms have been proposed in order to learn and model the user interests. The most well-known approaches are collaborative filtering (CF) and content-based (CB) algorithms, which are usually combined, creating hybrid RS. However, given their pervasiveness in the literature, we will skip their definitions here and refer the reader towards [5] to learn about CB systems and [12, 20] to know more about CF techniques, such as nearest neighbors and matrix factorization approaches. Hence, in this section we first focus on the specific aspects of venue recommendation (Section 2.1) and later we present the main concepts about cross-domain recommendation (Section 2.2).

2.1 Venue recommendation

As already mentioned, venue suggestion is related to the traditional recommendation problem, even though it exhibits some distinctive features. Because of this, we introduce now some specific notation for this task, following the one presented in [17]. Let us denote the set of users with \mathcal{U} and the set of venues (or POIs) with \mathcal{L} . We use \mathcal{L}^u to indicate the venues visited by user u in the training set. The main objective is to recommend venues that the user has not seen beforehand, even though repetitions can be exploited when building the recommendation model.

This type of recommendation is highly influenced by spatial, social, and temporal factors [17], so most of the algorithms try to model them in order to perform recommendations. Matrix factorization (MF) techniques are one of the most extended approaches in the RS area because they tend to outperform other techniques

such as neighbor-based algorithms [13], they can also be easily extended with baseline predictors and temporal information [12]. In venue recommendation they are also very popular and they are the core of many approaches, such as LRT [7], IReNMF [18], and others [14, 16]. As in classical recommendation, the basic operation in these algorithms is the minimization of the following function:

$$\min_{\mathcal{U}, \mathcal{L}} \|C - \hat{C}\|_F^2 + \lambda_1 \|\mathcal{U}\|_F^2 + \lambda_2 \|\mathcal{L}\|_F^2 \quad (1)$$

where \hat{C} is computed by the product of the user and POI matrices $\hat{C} = \mathcal{U}\mathcal{L}^T$, F is the Frobenius norm, and λ_1 and λ_2 are the regularization parameters. This is the basic algorithm, but it has been extended and adapted to this task. For example, the IReNMF approach [18] adds the geographical influence by using the k nearest neighbors of the target venue in the score function, where a clustering based on the POIs locations is also applied. The LRT algorithm [7] includes temporal information in the MF algorithm by introducing temporal steps to factorize the check-in matrix by each time slot; it then aggregates the result and adds a regularization term based on the similarities between the check-in matrix at different points in time. Other techniques using MF approximations at the core of their algorithms include the GeoMF [16] and the Rank-GeoFM [14] approaches.

However, other methods besides those based on MF have also been applied in venue recommendation. For example, the USG model from [28] combines three different components: a probabilistic model based on the history of visited POIs by the user to consider geographical information, CF similarities based on other users in the system (classic CF), and CF similarities based on the friends of the target user (social influence). Another interesting approach is the LORE algorithm from [31], where Markov Chains are used to model the sequential patterns between POIs, together with social and geographical influence. As can be seen, there is a great number of venue recommendation algorithms but the basis for most of them are proposals already explored in the traditional recommendation problem, complementing them with specific features of LBSNs (geographical, social, and temporal influence). In this paper, we propose to exploit cross-domain strategies in this context; in the next section we define the cross-domain problem and explain its main concepts.

2.2 Cross-domain recommendation

In the context of cross-domain recommendation, there exist multiple definitions of “domain” [3]: one can consider two items belong to different domains if they have different values for a specific attribute but others may argue different domains implies two separate systems or two types of items. In any case, the basic idea behind this type of recommendation is that, to improve recommendations over a target domain \mathcal{D}_T , some kind of knowledge from a source domain \mathcal{D}_S needs to be exploited. In order to illustrate these concepts, consider for example a source domain of books and a target domain of movies. We can make recommendations of users who do not have a mature rating history in the target domain by using some information stored in the source domain, i.e., if the target user liked books of drama and terror, she would probably also like movies of the same style even if the items are from different domains.

Depending on the information analyzed and the destination users to make recommendations, in [3] the following three main recommendation goals are described:

- Linked-domain recommendations: recommend items in the target domain by analyzing both the target and source domains.
- Cross-domain recommendations: recommend items in the target domain to users in the source domain by using only the information of the source domain.
- Multi-domain recommendations: recommend items belonging to the target or the source domain to all kind of users.

Furthermore, we must consider that \mathcal{D}_S and \mathcal{D}_T may share some information about the users, the items, or both, which allows us to categorize the different scenarios according to the data overlap as: no overlap, item overlap, user overlap, and full overlap (both user and item overlap). Additionally, we can also classify the cross-domain techniques according to how they exploit the knowledge. Based on [3], there are two different categories: “aggregating knowledge”, when the knowledge of the domains (user preferences, similarities, or single-domain recommendations) is aggregated, and “linking and transferring knowledge”, where there is a knowledge transfer between the different domains in order to produce recommendations, for instance by using common knowledge (semantic networks, items attributes, etc), sharing latent features, or transferring the rating patterns.

In Section 6 we survey some proposals where cross-domain techniques were used for venue recommendation and related tasks. Nonetheless, we want to emphasize we have not found many examples of cross-domain experiments combining more than two or three domains – usually, movies, music, and books –, except for the works presented in [21] and [23]. In the first one, the authors exploited the information of 10 domains (categories of different products from Epinions, hence, not related with tourism) in order to analyze the performance of the recommendations using ranking metrics (nDCG and Recall), however, these datasets are much smaller in terms of ratings than the ones we use in this paper (the largest one contains around 200K ratings) and they are all more dense; while in the second one, the authors used 21 domains (defined as the categories from different Yelp businesses) and measured the effects of cross-domain recommendation in terms of RMSE by comparing the performance on several domain pairs. Therefore, our work is – to best of our knowledge – the first study where each city defines a separate domain to exploit cross-domain strategies, using up to 8 domains to be combined into the source domain, and in the context of a tourism dataset with a realistic, temporal ranking-based evaluation.

3 ADAPTING CROSS-DOMAIN RECOMMENDATION TO VENUE SUGGESTION

Taking into account the characteristics of cross-domain recommendation, we propose to apply and adapt these concepts to venue recommendation. In LBSNs, it is natural to group the check-ins provided by the users per city, this, together with the issue that there is no venue overlap between different cities, has caused that

many researchers have isolated the check-ins of different cities and considered them as independent datasets [14, 15, 18].

In this paper, we consider each city as a different domain and propose to adapt strategies from cross-domain recommendation to combine these domains. The main advantage of applying some kind of cross-domain in venue recommendation is that we can expand the knowledge of the recommenders with a larger number of users and items, in order to establish more relationships between them. However, it is not obvious how the knowledge to be included should be selected, since noise might be added to the model or, simply, that information may not be useful at all.

The main goal we aim to achieve, if we focus on CF algorithms, would be to find highly active users in the aggregated domains so that they could alleviate the inherent sparsity problem. Therefore, we consider the following possibilities:

- Nearest cross-domain (N-CD): we use the n closest cities to the target city as the source domain. We aim to capture cultural patterns [27], while, at the same time, we keep control of the number of cities we consider.
- Most-popular cross-domain (P-CD): we use the n cities with more ratings as the source domain. This strategy allows us to test whether considering those cities that the system has more information about can be useful for the model. We hypothesize that having more information should be helpful for the recommendation algorithms, however, this strategy might also be more sensitive to noise and may not improve the user overlap.

The rationale behind our proposal is exposed in Table 1. Here we show the average of the percentage of common users between a given target city against the other cities that belong to the source domain, i.e., either the closest cities for N-CD or most popular cities for P-CD. Using the same dataset that will be used (and further explained) later in the experiments (prior partitioning and other processing), we compute the percentage of common users as follows (an analysis consistent with the one presented in [27]):

$$\text{Common Users}(C_1, C_2) = \frac{|\mathcal{U}(C_1) \cap \mathcal{U}(C_2)|}{|\mathcal{U}(C_1) \cup \mathcal{U}(C_2)|} \quad (2)$$

where $\mathcal{U}(C)$ denotes the set of users in city C . We observe that the N-CD strategy is really useful to find more users in common, however, it is not clear the actual effect this characteristic may have on the performance of CF algorithms. At the same time, even though the P-CD strategy does not discover many users in common, since it includes much more data to train the recommenders, it may be more beneficial to some recommendation approaches.

More formally, the knowledge exploitation between cities that we propose can be done with both knowledge aggregation or linking knowledge, as we can aggregate all the cities in the source domain to make recommendations but we can also transfer knowledge (e.g., with attributes, latent features, semantic networks) between them. Thus, we adapt the optimization formula from [1] to include both venue and cross-domain recommendation as follows:

$$\forall u \in \mathcal{U}, l'_u = \arg \max_{l \in \mathcal{L}^{\mathcal{D}_T}} g(u^{\mathcal{D}_T}, l^{\mathcal{D}_T}; \mathcal{U}^{\mathcal{D}_S}, \mathcal{L}^{\mathcal{D}_S}) \quad (3)$$

where \mathcal{D}_S and \mathcal{D}_T are the source and the target domains respectively. Hence, we aim to maximize the score that a user has for an

Table 1: Average percentage of common users with respect to the other cities included in each cross-domain strategy.

Cross-domain Strategy	Cities							
	Istanbul	Jakarta	Kuala Lumpur	Mexico City	Moscow	Santiago	São Paulo	Tokyo
N-CD	9.36%	13.58%	28.77%	11.61%	2.82%	13.07%	7.43%	33.26%
P-CD	0.33%	1.10%	0.90%	0.61%	0.36%	0.74%	0.62%	0.49%

item (both in the target domain) by using information about users and items in the source domain.

Taking into account the characteristics shown in Section 2.2, our cross-domain proposal satisfies:

- Linked-domain recommendation, where we recommend items in the target domain (a specific city) by exploiting the knowledge derived from the source (data from other cities) and target domains.
- Scenario with only user overlap, since an item (venue) from one domain will never appear in a different domain.
- Aggregated knowledge (merging user preferences), since we combine multiple sources of personal preferences (basically, the check-ins from various cities and the target city).

4 EXPERIMENTAL SETUP

4.1 Dataset

The experiments have been performed using the global-scale check-in dataset of Foursquare¹ made public by the authors of [26, 27]. Starting from more than 33M check-ins, we created one temporal split containing 6 months of data in its training split and one month for testing, more statistics are shown in Table 2. As a pre-processing step, we performed a 2-core before splitting the data into training and test, so that every user and item has at least 2 preferences. Furthermore, in order to make a fair comparison among all the evaluated baselines, we decided to remove repetitions in a user basis, even though some algorithms, such as IReMF, are able to exploit the frequency of users when visiting a specific venue.

Additionally, the time of the check-in in the original dataset was in UTC with the offset in minutes; hence, the local time is the UTC time plus the offset. However, we decided to work with the local date of each check-in in order to select the training and test subsets independently of where the check-in took place, as it is more realistic if the splits are made on the same local day for all cities, and not with respect to the global UTC schedule.

4.2 Methods for comparison

We test the following state-of-the-art recommenders:

- Random (Rnd): random recommender.
- Popularity (Pop): recommender that suggests the most popular items, i.e., items with more check-ins.
- AvgDis: baseline recommender that recommends the closest POIs to the user's average location. The average is computed by calculating the midpoint of the coordinates of the POIs visited by each user.
- PGN: a hybrid approach similar to the USG model proposed in [28] that combines Pop, UB, and AvgDis recommenders. It basically aggregates the scores of every item provided by

each of the recommenders, after normalizing each score by the maximum score of each method.

- UB: a k -NN recommender with a user-based approach [20].
- IB: a k -NN recommender with an item-based approach [20].
- HKV: a matrix factorization (MF) approach as in [9] that uses Alternate Least Squares in the minimization formula.
- IReMF: weighted MF method proposed by [18] and briefly described in Section 2.1. We selected this approach because, according to the comparison presented in [17], IReMF was very competitive with a lower execution time with respect to other models, such as GeoMF [16], Rank-GeoFM [14], or LFB-BCA [25], which agrees with some preliminary experiments we performed in our dataset.

The parameters tested for these algorithms are detailed in Table 3, where 83 different configurations were tested in total for each city. The final parameters are shown in Table 4, and they were found by optimizing P@5 for the single-domain scenario. For every recommender except the IReMF algorithm, we used the RankSys library [4]; for IReMF we used the implementation provided by [18], available here². Source code to replicate these experiments can be found in the following Bitbucket repository: PabloSanchezP/TempCDSeqEval.

4.3 Evaluation methodology

Based on the temporal split presented in Table 2, we decided to focus on the eight largest cities in terms of number of check-ins: Mexico City, São Paulo, Moscow, Kuala Lumpur, Santiago, Tokyo, Jakarta, and Istanbul.

To evaluate the recommenders, we applied the TrainItems methodology [24], where only the venues that appear in the training set of each target city are considered as candidates, except the ones already rated by the user. Then, according to what we present in Section 3, we compare the recommendations generated by the different algorithms when using different training information: either only the target city (single-domain, or SD), or a combination of cities (cross-domain, or CD) where we explore two possibilities – nearest-based CD (N-CD) and most popular CD (P-CD). Unless stated otherwise, the reported values are computed at a cutoff of 5.

5 RESULTS

In this section, we present first the results obtained when considering each city as a separate domain (Section 5.1) and then we analyze how different cross-domain strategies affect the performance of single-domain scenarios (Section 5.2).

5.1 Analysis of single-domain performance

We show in Table 5 the results for the recommenders presented in Section 4.2 in terms of nDCG@5 for the single-domain scenario. The

¹<https://sites.google.com/site/yangdingqi/home/foursquare-dataset>

²<http://spatialkeyword.sce.ntu.edu.sg/eval-vldb17/>

Table 2: Description of the temporal partition evaluated created based on the Foursquare dataset, where U , I , and C denote the number of users, items, and check-ins.

Check-in period	U	I	C	Density	C/U	C/I
Apr'12-Sep'13	267k	3.6M	33M	0.0034%	123.596	9.16
Training: May-Oct '12	202k	1.1M	4.7M	0.0021%	23.267	4.278
Test: Nov '12	150k	352k	831k	0.0017%	5.540	2.361

Table 3: Parameters used with the evaluated recommenders. SC and SJ stand for SetCosine (as proposed in [2], with $\alpha = 0.5$) and SetJaccard, i.e., implicit versions of the well-known similarity metrics.

Recommender	Parameters
Rnd	None
Pop	None
AvgDis	None
PGN	$k = 100$, Similarity = SJ
UB	Similarity = {SC, SJ}, $k = \{5, 10, \dots, 100\}$
IB	Similarity = {SC, SJ}, $k = \{5, 10, \dots, 100\}$
HKV	Factors = {10, 50, 100}, $\alpha = \{0.1, 1, 10\}$, $\lambda = \{0.1, 1, 10\}$
IRenMF	$k = 10$, Clusters = 50, $\lambda_1 = \lambda_2 = 0.015$, Factors = {50, 100}, $\alpha = \{0.4, 0.6\}$, $\lambda_3 = \{0.1, 1\}$

Table 4: Optimal parameters of recommenders for each city. The order of the presented parameters is: for UB and IB, similarity and neighborhood size; for HKV, factors, α , λ ; for IRenMF, factors, α , λ_3 .

City \ Rec	UB	IB	HKV	IRenMF
Istanbul (IST)	SJ, 90	SC, 100	10, 10, 10	100, 0.4, 1
Jakarta (JAK)	SJ, 100	SC, 80	10, 10, 10	50, 0.4, 1
Kuala Lumpur (KUA)	SJ, 100	SJ, 100	10, 10, 10	100, 0.4, 0.1
Mexico City (MEX)	SJ, 100	SJ, 100	10, 10, 10	50, 0.4, 1
Moscow (MOS)	SC, 100	SJ, 100	50, 10, 1	100, 0.4, 1
Santiago (SAN)	SJ, 90	SJ, 80	10, 10, 10	100, 0.4, 1
São Paulo (SAO)	SJ, 100	SJ, 100	50, 10, 0.1	100, 0.4, 0.1
Tokyo (TOK)	SJ, 80	SC, 80	10, 10, 10	100, 0.4, 1

first thing we note is the low values obtained by the recommenders. This is mostly due to the high sparsity of the data (see Table 2), together with the fact that we are using a temporal split, which makes the recommendation task even more difficult, since, for instance, some of the few items a user may have in her test set may not appear in the training set at all, making such recommendation impossible to be produced by a collaborative algorithm.

Moreover, the AvgDis recommender is the second worst algorithm (after Rnd), which could be attributed to how we processed the data: since each venue visited by the user has now the same importance (because we removed repetitions), the user's center

cannot consider the most frequent venues in their computation and, hence, it might not represent her preferred areas in the city.

Similarly, we observe that the IRenMF approach, even though it remains very competitive, it never emerges as the optimal recommender. This might be caused by several reasons; first, as with the AvgDis recommender, it is able to work with repeated values and without them it loses some efficiency; secondly, its claimed superior performance was only tested using a random split in [18] instead of a temporal evaluation; and, finally, classical recommendation algorithms such as Pop or standard CF approaches were neglected in [17] which, together with our previous discussion, definitely disturbs such comparisons.

For the rest of the recommenders, we observe that UB is one of the best approaches for most of the cities. This result might be misleading as this recommender actually has less user coverage than other approaches [8]. It is interesting to note the relatively high performance of PGN, since it is able to beat the rest of the baselines in many cities, despite its simplicity and the fact that we did not perform any parameter tuning.

5.2 Performance comparison of cross-domain strategies

Table 6 shows the results for all cities with two different cross-domain approaches: P-CD (for every city, the training set is built by combining the training data from the eight selected cities) and N-CD (the training set is built by taking the nearest 7 cities with respect to the target city, so the number of cities under consideration is comparable to that of P-CD). In the same table, we show the improvement with respect to the single-domain scenario for every algorithm except Rnd and Pop, since they do not change their recommendations when changing the training set and keeping the same test set.

We observe that the performance improvement when using the P-CD strategy is usually negligible, except for TOK and HKV; in general, most of the improvements when using this strategy are very close to zero and, for many of the city-recommender combinations, extremely negative. This is especially true for the MF approaches (HKV and IRenMF), which seems to indicate that having more data available so that the sparsity is reduced does not guarantee better recommendations. Moreover, N-CD usually produces larger improvements with less training data involved (since the nearest cities always include less check-ins than the originally selected cities, which were the most popular ones in our dataset). This seems to confirm that better data is more useful than more data, in particular, the amount of user overlap is a good signal of the impact that a cross-domain strategy may obtain (as evidenced by the data presented before in Table 1).

Table 5: Performance results for the single-domain scenario in terms of nDCG@5. In bold, we show the highest value for each city.

City	Rnd	Pop	AvgDis	PGN	UB	IB	HKV	IRenMF
IST	0.000	0.054	0.001	0.067	0.073	0.059	0.070	0.069
JAK	0.000	0.066	0.001	0.067	0.070	0.035	0.066	0.065
KUA	0.000	0.066	0.001	0.070	0.073	0.042	0.066	0.070
MEX	0.000	0.041	0.001	0.043	0.044	0.013	0.047	0.043
MOS	0.000	0.027	0.002	0.032	0.037	0.017	0.039	0.035
SAN	0.000	0.051	0.001	0.054	0.053	0.026	0.050	0.052
SAO	0.000	0.053	0.001	0.057	0.049	0.015	0.048	0.043
TOK	0.000	0.069	0.001	0.070	0.069	0.048	0.059	0.068

Table 6: Performance results for the cross-domain scenario in terms of nDCG@5: N-CD and P-CD indicate a cross-domain strategy using the nearest 7 cities and the most popular 7 cities, respectively. The improvement in performance with respect to the single-domain scenario (Table 5) is represented as $\Delta(\%)$, where \blacktriangle (\blacktriangledown) denotes the largest positive (negative) improvement.

City		AvgDis	PGN	UB	IB	HKV	IRenMF
IST	N-CD	0.001	0.068	0.073	0.057	0.071	0.059
	$\Delta(\%)$	-9.7	1.6	0.3	-3.2	\blacktriangle 2.0	\blacktriangledown -14.8
	P-CD	0.001	0.068	0.073	0.059	0.068	0.052
	$\Delta(\%)$	-0.1	\blacktriangle 0.9	0.4	0.0	-3.4	\blacktriangledown -24.7
JAK	N-CD	0.001	0.070	0.075	0.034	0.070	0.062
	$\Delta(\%)$	\blacktriangledown -17.3	4.8	\blacktriangle 6.7	-3.9	6.3	-4.5
	P-CD	0.001	0.068	0.071	0.035	0.060	0.058
	$\Delta(\%)$	-9.9	\blacktriangle 0.6	0.4	-0.8	-8.5	\blacktriangledown -10.7
KUA	N-CD	0.001	0.072	0.076	0.040	0.075	0.069
	$\Delta(\%)$	\blacktriangledown -36.3	2.1	4.1	-3.9	\blacktriangle 13.8	-1.0
	P-CD	0.001	0.070	0.073	0.042	0.065	0.064
	$\Delta(\%)$	-0.8	0.0	\blacktriangle 0.3	-0.3	-1.6	\blacktriangledown -8.8
MEX	N-CD	0.001	0.044	0.045	0.013	0.045	0.040
	$\Delta(\%)$	\blacktriangle 13.3	2.2	1.6	-6.5	-5.0	\blacktriangledown -6.8
	P-CD	0.001	0.044	0.045	0.013	0.037	0.037
	$\Delta(\%)$	-0.2	\blacktriangle 1.3	1.2	-0.1	\blacktriangledown -22.1	-13.6
MOS	N-CD	0.002	0.033	0.038	0.017	0.040	0.034
	$\Delta(\%)$	\blacktriangledown -6.9	0.8	2.5	-0.7	\blacktriangle 3.3	-1.1
	P-CD	0.002	0.032	0.037	0.018	0.036	0.029
	$\Delta(\%)$	-0.6	0.1	0.3	\blacktriangle 1.1	-7.7	\blacktriangledown -17.4
SAN	N-CD	0.001	0.059	0.060	0.025	0.060	0.055
	$\Delta(\%)$	8.0	8.7	13.0	\blacktriangledown -5.0	\blacktriangle 20.2	4.9
	P-CD	0.001	0.055	0.054	0.026	0.046	0.045
	$\Delta(\%)$	-0.7	\blacktriangle 1.1	0.9	-0.3	-7.9	\blacktriangledown -13.2
SAO	N-CD	0.001	0.057	0.056	0.016	0.056	0.046
	$\Delta(\%)$	\blacktriangledown -7.1	0.4	\blacktriangle 15.4	5.5	15.2	7.3
	P-CD	0.001	0.057	0.049	0.015	0.047	0.034
	$\Delta(\%)$	-9.2	\blacktriangle 0.5	-0.2	-0.2	-2.1	\blacktriangledown -20.2
TOK	N-CD	0.000	0.073	0.073	0.048	0.064	0.071
	$\Delta(\%)$	\blacktriangledown -15.6	4.9	5.4	-0.2	\blacktriangle 8.7	4.2
	P-CD	0.001	0.070	0.069	0.048	0.064	0.064
	$\Delta(\%)$	-0.3	-0.2	-0.2	-0.1	\blacktriangle 8.6	\blacktriangledown -6.1

Classical CF algorithms like UB and HKV are able to exploit more successfully the information coming from the source domains, except, as discussed before, HKV with P-CD; the IB approach, on the other hand, obtained in general very poor results. We argue this trend for the CF techniques is related to whether the new users – recall that the new items will never have overlap with the target items, since this type of cross-domain has only user overlap by definition – have some kind of interaction with the target city. In this sense, when combining information from nearby cities it is more likely to find similar users with useful suggestions or learning relevant latent representations more related to the target items.

On the other hand, cross-domain strategies tend to deteriorate the performance of the techniques based on geographical distances (AvgDis and IRenMF). The reason for this might be quite obvious, since considering other cities to compute a new user's centroid will certainly move such centroid far away from the target city, which is not useful when we are only interested in recommending venues inside of that specific city.

Therefore, we are able to answer our two research questions. First, regarding **RQ1** (Are state-of-the-art recommenders able to exploit cross-domain information in the context of venue recommendation?) we have seen that some classic recommenders are able to benefit from cross-domain information if the cities are selected properly, however, other approaches more tailored for venue recommendation (such as those exploiting geographical information) seem to decrease their performance when exploiting knowledge from other domains. Second, as an answer to **RQ2** (Which is the best cross-domain strategy in terms of relevance?), we can conclude that selecting cities by proximity (the closest ones) has a greater benefit than selecting them by the amount of information they contain (popularity).

5.3 Discussion

According to the results obtained, applying cross-domain techniques can improve the results obtained in some situations, although their effect is not as great as one might expect (mainly due to the temporal split we applied and the sparsity of the dataset). However, since we mostly explored very basic recommendation techniques, these results are very promising. First of all, because we have seen that some of the algorithms are able to make better recommendations (in some cases, up to a 20% improvement) – however, further analysis should be done to understand the impact of such improvement in other evaluation dimensions, such as novelty or diversity. Secondly, due to the well-known popularity bias [11], such a simple technique could outperform other methods like IB or AvgDis (see Table 5), even though this type of baseline is usually ignored in POI recommendation literature. Finally, a negative result we observed is that when the distance between the venues is considered, applying cross-domain can be counterproductive and is not recommended.

It should be noted that our results are consistent with those discussed in [23], where the authors found that there are specific experiments in which cross-domain recommendation works worse than classic single-domain recommendations, even though in general it behaves better or as good as single-domain strategies. Nonetheless, only comparisons between single-domain and cross-domain

approaches on three different algorithms and without considering any temporal split were presented in that paper; hence, our work helps on generalizing the conclusions obtained in such paper.

We want to emphasize that considering knowledge from different cities (understood as different domains), despite being computationally more expensive, has a clear advantage: such system would only need to train once for each city where recommendations are required, whereas considering each city as an isolated training domain (called single-domain in this paper) only allows to generate recommendations for a single city; hence, the recommendation model built can be re-used more often in the former case, at the expense of being more expensive in terms of memory and time consuming. However, as we have shown here, if the cross-domain model is generated based on the *right* cities, significant performance improvements can be achieved, not always by selecting the cities with more information but those that are closer and more likely to have overlap in their users, probably because they are culturally related and share similar mobility patterns [27].

6 RELATED WORK

Cross-domain techniques can help recommendation systems in a number of situations. Specifically, travel recommendation was addressed as a potential target of these techniques [33]. There have been some papers in which researchers explored different ways to combine sources of information to be applied in tourism. For example, in [22] the authors describe TourMIS, a dataset of European statistics of tourism data, where they show the usefulness of combining different data sources (economy, tourism, and sustainability) to make relations between them. Although they indicate that sometimes it is difficult to integrate sources from very different domains, most of those problems can be solved using Linked Data approaches. Nevertheless, the dataset described was not used to produce venue recommendations, only for statistical analysis. Another approach of cross-domain in tourism can be seen in [6], where the authors did not recommend POIs but music artists depending on the monument the user is visiting; they do this by building a semantic network of venues and artists and the relations between them. In [32], the authors exploited information from different sources like location-activity, location-feature, and activity-activity correlations to enhance the performance, showing a 20% improvement over a basic algorithm that does not use any additional data.

However, the most similar approach to this paper that we found is [29]. In that article, the authors perform a so-called cross-region recommendation, and considered each region as a different domain. One important difference with our work is that, whereas a division by cities is natural, they used regions computed by performing clustering on the venues. Additionally, the datasets used (Foursquare and Yelp) are smaller than the one reported here and no temporal evaluation was performed, only a standard cross validation methodology. Hence, our paper offers a complementary view on a related problem, from a more realistic perspective (since we explicitly address a time-aware evaluation) with a larger dataset and taking into account the check-ins in different cities to perform the recommendations.

7 CONCLUSIONS AND FUTURE WORK

In this paper we have explored venue recommendation with a novel approach, by applying concepts of cross-domain recommendation under a realistic scenario, by using a temporal evaluation methodology. We have shown an empirical evaluation comparing the performance of state-of-the-art recommenders under different settings (single-domain and cross-domain applied on the closest cities and the most popular ones). Even though the behavior varies in every city, the cross-domain strategy based on the closest cities tends to produce better results; in the future, we would like to exploit this observation to create a generic recommender system for venue suggestion that takes this information into account, since so far in this work we have not proposed any new algorithm using cross-domain data.

In any case, we believe there is still room for improvement. First of all, the evaluated algorithms are very simple, this study should probably be repeated considering more complex methods such as [19] or [7]. Nevertheless, POI recommendation algorithms that exploit geographic information may be negatively affected by the cross-domain strategies as user movement patterns may be modified if we use information from other cities checkins. On the other hand, the temporal evaluation methodology should also be analyzed more carefully, especially regarding the effect of seasonal trends and how it may affect the knowledge transfer techniques (since not enough interactions or users might be available or active at the same time).

We aim to further extend how the cities are selected by computing some kind of similarity between them using content and cultural information [27]. We would also like to exploit the venue categories, social connections, or other content information, and see how that information is affected when using cross-domain strategies. Furthermore, due to the high sparsity of this recommendation task, we believe that a study to analyze the cold-start problem is needed in order to check if cross-domain strategies perform well under these circumstances. Even though in this paper we performed a 2-core pre-processing and this setting is still realistic enough for several situations, a more thorough analysis should be performed to understand the sensitivity of this problem to the different algorithms and cross-domain strategies.

ACKNOWLEDGMENTS

This work was funded by the research project TIN2016-80630-P (MINECO).

REFERENCES

- [1] Gediminas Adomavicius and Alexander Tuzhilin. 2005. Toward the Next Generation of Recommender Systems: A Survey of the State-of-the-Art and Possible Extensions. *IEEE Trans. Knowl. Data Eng.* 17, 6 (2005), 734–749.
- [2] Fabio Aiolli. 2013. Efficient top-n recommendation for very large scale binary rated datasets. In *RecSys*. ACM, 273–280.
- [3] Iván Cantador, Ignacio Fernández-Tobías, Shlomo Berkovsky, and Paolo Cremonesi. 2015. Cross-Domain Recommender Systems. In *Recommender Systems Handbook*. Springer, 919–959.
- [4] Pablo Castells, Neil J. Hurley, and Saul Vargas. 2015. Novelty and Diversity in Recommender Systems. In *Recommender Systems Handbook*. Springer, 881–918.
- [5] Marco de Gemmis, Pasquale Lops, Cataldo Musto, Fedelucio Narducci, and Giovanni Semeraro. 2015. Semantics-Aware Content-Based Recommender Systems. In *Recommender Systems Handbook*. Springer, 119–159.
- [6] Ignacio Fernández-Tobías, Iván Cantador, Marius Kaminskas, and Francesco Ricci. 2011. A Generic Semantic-based Framework for Cross-domain Recommendation. In *Proceedings of the 2Nd International Workshop on Information Heterogeneity and Fusion in Recommender Systems (HetRec '11)*. ACM, 25–32.
- [7] Huiji Gao, Jiliang Tang, Xia Hu, and Huan Liu. 2013. Exploring temporal effects for location recommendation on location-based social networks. In *RecSys*. ACM, 93–100.
- [8] Asela Gunawardana and Guy Shani. 2015. Evaluating Recommender Systems. In *Recommender Systems Handbook*. Springer, 265–308.
- [9] Yifan Hu, Yehuda Koren, and Chris Volinsky. 2008. Collaborative Filtering for Implicit Feedback Datasets. In *ICDM*. IEEE Computer Society, 263–272.
- [10] Dietmar Jannach, Lukas Lerche, Iman Kamehkhosh, and Michael Jugovac. 2015. What recommenders recommend: an analysis of recommendation biases and possible countermeasures. *User Model. Interact.* 25, 5 (2015), 427–491.
- [11] Yehuda Koren and Robert M. Bell. 2015. Advances in Collaborative Filtering. In *Recommender Systems Handbook*. Springer, 77–118.
- [12] Yehuda Koren, Robert M. Bell, and Chris Volinsky. 2009. Matrix Factorization Techniques for Recommender Systems. *IEEE Computer* 42, 8 (2009), 30–37.
- [13] 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 *SIGIR*. ACM, 433–442.
- [14] Xin Li, Mingming Jiang, Huiting Hong, and Lejian Liao. 2017. A Time-Aware Personalized Point-of-Interest Recommendation via High-Order Tensor Factorization. *ACM Trans. Inf. Syst.* 35, 4 (2017), 31:1–31:23.
- [15] Defu Lian, Cong Zhao, Xing Xie, Guangzhong Sun, Enhong Chen, and Yong Rui. 2014. GeoMF: joint geographical modeling and matrix factorization for point-of-interest recommendation. In *KDD*. ACM, 831–840.
- [16] Yiding Liu, Tuan-Anh Pham, Gao Cong, and Quan Yuan. 2017. An Experimental Evaluation of Point-of-interest Recommendation in Location-based Social Networks. *PVLDB* 10, 10 (2017), 1010–1021.
- [17] Yong Liu, Wei Wei, Aixin Sun, and Chunyan Miao. 2014. Exploiting Geographical Neighborhood Characteristics for Location Recommendation. In *CIKM*. ACM, 739–748.
- [18] Jarana Manotumruksa, Craig Macdonald, and Iadh Ounis. 2017. A Deep Recurrent Collaborative Filtering Framework for Venue Recommendation. In *CIKM*. ACM, 1429–1438.
- [19] Harvey J. Miller. 2004. Tobler's First Law and Spatial Analysis. *Annals of the Association of American Geographers* 94, 2 (2004), 284–289. <https://doi.org/10.1111/j.1467-8306.2004.09402005.x>
- [20] Xia Ning, Christian Desrosiers, and George Karypis. 2015. A Comprehensive Survey of Neighborhood-Based Recommendation Methods. In *Recommender Systems Handbook*. Springer, 37–76.
- [21] Dimitrios Rafailidis and Fabio Crestani. 2017. A Collaborative Ranking Model for Cross-Domain Recommendations. In *CIKM*. ACM, 2263–2266.
- [22] Marta Sabou, Irem Onder, Adrian M. P. Brasoveanu, and Arno Scharl. 2016. Towards cross-domain data analytics in tourism: a linked data based approach. *J. of IT & Tourism* 16, 1 (2016), 71–101.
- [23] Shaghayegh Sahebi and Peter Brusilovsky. 2015. It Takes Two to Tango: An Exploration of Domain Pairs for Cross-Domain Collaborative Filtering. In *RecSys*. ACM, 131–138.
- [24] Alan Said and Alejandro Bellogín. 2014. Comparative recommender system evaluation: benchmarking recommendation frameworks. In *RecSys*. ACM, 129–136.
- [25] Hao Wang, Manolis Terrovitis, and Nikos Mamoulis. 2013. Location recommendation in location-based social networks using user check-in data. In *SIGSPATIAL/GIS*. ACM, 364–373.
- [26] Dingqi Yang, Daqing Zhang, Longbiao Chen, and Bingqing Qu. 2015. NationTelescope: Monitoring and visualizing large-scale collective behavior in LBSNs. *J. Network and Computer Applications* 55 (2015), 170–180.
- [27] 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 (2016), 30:1–30:23.
- [28] Mao Ye, Peifeng Yin, Wang-Chien Lee, and Dik Lun Lee. 2011. Exploiting geographical influence for collaborative point-of-interest recommendation. In *SIGIR*. ACM, 325–334.
- [29] Chenyi Zhang and Ke Wang. 2016. POI recommendation through cross-region collaborative filtering. *Knowl. Inf. Syst.* 46, 2 (2016), 369–387.
- [30] Jia-Dong Zhang and Chi-Yin Chow. 2013. iGSLR: personalized geo-social location recommendation: a kernel density estimation approach. In *SIGSPATIAL/GIS*. ACM, 324–333.
- [31] Jia-Dong Zhang, Chi-Yin Chow, and Yanhua Li. 2014. LORE: exploiting sequential influence for location recommendations. In *SIGSPATIAL/GIS*. ACM, 103–112.
- [32] Vincent Wenchen Zheng, Yu Zheng, Xing Xie, and Qiang Yang. 2010. Collaborative location and activity recommendations with GPS history data. In *WWW*. ACM, 1029–1038.
- [33] Yu Zheng. 2015. Methodologies for Cross-Domain Data Fusion: An Overview. *IEEE Trans. Big Data* 1, 1 (2015), 16–34.