Towards a Reconciliation of Point-of-Interest Recommendation Systems Using Multi-Agent Systems

Zoulikha Maghni Sandid[1,3,*], Zohra Slama[1] and Nassim Dennouni[1,2]

[1]*EEDIS Laboratory, Djillali Liabes University of Sidi Bel Abbes, Algeria*
[2]*Higher School of Management, Tlemcen, Algeria*
[3]*Abdelhamid Ben Badis University of Mostaganem, Algeria*
E-mail: maghnisandid.z@yahoo.com; zoulikha.maghni-sandid@univ-sba.dz; zslama@yahoo.com; dennouninas@gmail.com
[*]*Corresponding Author*

Received 22 January 2025; Accepted 04 July 2026

Abstract

Point-of-interest (POI) recommendation is a good way to help Smartphone users discover new places by exploiting their preferences and social networks. However, this technique, which is often based on collaborative filtering, suffers from cold-start problem due to insufficient interaction with POIs (ratings) and the absence of declared friendship or trust relationships between users. To address this problem, one solution is to launch several Recommendations Systems (RS) simultaneously, using all available sources of information within the social network, especially in the case of a new user or a new POI. In this paper, a Multi-Agent System for Reconciling POI Recommendation Algorithms (MSRPRA) is proposed to exploit (1) the power of Pearson similarity deduced from POI ratings through the RatAg agent, (2) the effectiveness of Jaccard similarity derived from existing friendship relationships exploited by the FrAg agent and the contribution of trust scores declared

Journal of Mobile Multimedia, Vol. 22_4, 475–496.
doi: 10.13052/jmm1550-4646.2243

by users using the TrAg agent. The system then uses a coordinator agent to merge, using the Borda voting method, the POIs lists generated by the RatAg, FrAg, and TrAg agents. The experimental results show that MSRPRA outperforms approaches based solely on Pearson similarity, Jaccard similarity, or explicit trust. On average, the model improves Precision by 7.38% and Recall by 7.73%, while achieving better precision in terms of RMSE, these results confirm the effectiveness of multi-agent fusion using Borda voting in improving the relevance of recommendations and mitigating the cold-start problem.

Keywords: Recommendation systems, collaborative filtering, point-of-interest, cold-start problem, multi-agent system, reconciliation, Borda.

1 Introduction

Point-of-interest (POI) recommendation systems (RS) are essential for guiding users to locations of potential interest, such as restaurants, tourist attractions, or cultural events [1]. However, these systems can face limitations, particularly in terms of personalization, consideration of user context, and management of diverse information sources. For this reason, Artificial Intelligence (AI) algorithms, such as deep learning [2] and decision trees [3], have been integrated into various domains like e-tourism [4] and e-commerce [5], to exploit the potential of vast amounts of data collected during system usage, making them more intelligent.

Prior to their maturity, RS use two basic techniques [6]: (1) Collaborative Filtering Technique (CFT) and (2) Content-Based Filtering Technique (CBT). CFT identify similar tourists to predict POIs that should be visited [7], while CBTs analyze the characteristics of POIs to suggest those that match to the tourist's preferences or interests [8, 9]. In the context of POIs recommendation, it is challenging to accurately characterize each location due of the complexity of geographic information, mainly linked to its spatiotemporal nature [10]. For this reason, CFT are more appropriate to this situation, as they rely on the explicit evaluations of POIs provided by tourists to calculate similarities, which serve as the starting point for the recommendation process [11].

However, these systems suffer from the problem of data sparsity caused by user indifference and biased ratings, as most tourists do not place significant importance on rating the places they have already visited [12]. One solution to this issue is to combine several recommendation techniques running in parallel, leveraging other types of data such as trust and friendships.

To achieve this objective, we designed three agents: (a) RatAg that uses Pearson similarity based on ratings, (b) FrAg that uses Jaccard similarity based on friendships between users, and (c) TrAg based on trust. However, reconciling the POIs lists obtained from these three recommendation algorithms presents a significant challenge. To overcome this limit, the use of Multi-Agent Systems (MAS) enables an effective combination of recommendation algorithms to achieve intelligent reconciliation.

For these reasons, we propose a Multi-agent System for Reconciling POI Recommendation Algorithms (MSRPRA). This system consists of three layers. The first layer includes a context-aware agent, named ContextAg, responsible for collecting spatiotemporal contexts and the preferences. The second layer comprises the three recommendation agents (RatAg, FrAg, TrAg), each independently generating a ranked list of POIs. The third layer contains the reconciliation coordinator agent named CorAg, which reconciles and merges these three ranked lists and returns a comprehensive list of the most relevant POIs using the Borda voting technique [13]. This architecture combines independent recommendation generation with fusion, enabling the exploitation of complementary information sources and enhancing the overall robustness of the RS.

The primary contribution of this work is the proposal of a reconciliation strategy based on the Borda voting method derived from social choice theory, to merge the rankings generated by multiple recommendation agents. Unlike traditional aggregation techniques, the MSRPRA model produces a consensus raking by taking into account the relative position of each POI across all lists, ensuring a simple, robust aggregation that is particularly well-suited to sparse data and cold-start scenarios.

To address this issue, the MSRPRA system (see Figure 1) combines several sources of information, namely Pearson similarity, Jaccard similarity, and explicit trust relationships between users. The integration of this social information, combined with Borda aggregation mechanism, improves the quality of recommendations when user-POI interactions are limited.

The main contributions of this work are:

- Proposal of a multi-source recommendation framework that integrates Pearson similarity, Jaccard similarity, and explicit trust relationships.
- Introduction of a Borda voting-based aggregation method to reconcile the recommendation lists produced by multiple recommendations agents.

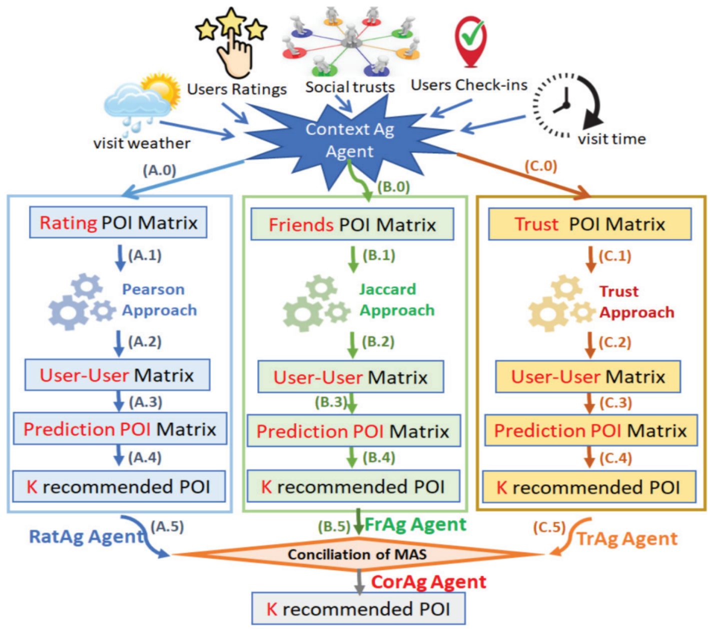

Figure 1 Functional architecture of our system.

- Experimental validation demonstrating promising performance in terms of RMSE, Precision, and Recall compared to approaches that use Pearson similarity, Jaccard similarity, or explicit confidence individually.
- Evaluation of the effectiveness of vote-based aggregation in mitigating the cold-start problem in POI RS.

The remainder of the paper is structured as follows. Section 2 presents a state-of-the-art review of approaches for developing RS using multi-agents systems. Section 3 describes our recommendation model. Section 4 details the experimentation phase, including an analysis and discussion of results. Section 5 provides a summary of the contributions and perspectives of our work.

2 Related Work

In recent years, several studies have focused on RS using MAS. These systems, known as Multi-Agent Recommendation Systems (MARS), are RS that integrate autonomous and cooperative agents to enhance the quality of recommendations [14].

Sebastia et al. [15] proposed a MARS for recommending tourist attractions in the city of Valencia (Spain) based on an activity plan. However, the system requires continuous user involvement throughout the recommendation process, as tourists must explicitly specify their preferences at each stage and provide feedback by rating the attractions they have visited.

In the same context, Bedi et al. [16] proposed a MARS for e-tourism that employs reputation-based collaborative filtering to recommend various services, including hotels, tourist attractions, and restaurants. In this framework, each type of service is handled by a dedicated agent responsible for generating its own recommendation list. Although the proposed approach effectively exploits the capabilities of MAS to distribute and coordinate recommendation tasks, it suffers from the issue of sparse matrix.

In another context, Morais et al. [17] proposed a multi-agent model for website personalization by combining two recommendation techniques, namely association rule mining and collaborative filtering. To improve system performance, they employed agents to reduce response time and separated the updates processing from their models. Furthermore, their approach adopts an incremental learning strategy, allowing the system to incorporate new information after each session. However, this incremental mechanism requires a significant number of updates, which may increase the computational overhead.

To address this type of problem, a centralized MAS architecture was proposed in [18] to recommend locations to users. The server agent collects data from client agents to calculate rating predictions. Subsequently, the list of the most relevant locations is sent to the client agent which, in turn, selects those within a specific location perimeter for display.

On the other hand, a video RS based on MAS was proposed in [19]. They propose seven agents (location, age, financial, identity, personality, needs, social) along with an Information Center Agent (ICA) that collects and processes the data. Each agent indicates whether the POI is interesting or not, and the ICA ranks the POIs accordingly. However, in this system, individual agents do not rank the video; they only express their interest or not towards the video.

In the same context, Kaur et al. [20] proposed a MARS aimed at improving user retention on the Netflix platform by recommending advertisements for movies and series that match users' preferences. This system employs a supervised learning algorithm based on K-nearest neighbors (KNN) with cosine similarity. User ratings for movies were used to calculate predictions. However, such platforms suffer from the data sparsity problem.

To assist learners, MARS are used as online learning platforms that are increasingly sought after by students due to the growing number of courses available online. For this reason, Amane et al. [21] have proposed a content-based RS using two agents. Their system operates in two stages: similarity calculation and reclassification through filtering. These two agents extract negative and positive comments in order to reclassify the top-k courses list.

To improve recommendation accuracy, Alhejaili and Fatima[22] utilized a multi-agent recommender system composed of six agents, five of which implement different machine learning techniques, namely Random Forest, Artificial Neural Networks, Support Vector Machines (SVM), KNN, and Naïve Bayes. The agents coordinate and negotiate using the Contract-Net protocol to produce the final recommendation. Although the framework benefits from the diversity of multiple learning algorithms, all recommendation agents process the same input data, limiting the diversity of information exploited by the system and reducing the potential advantages of the multi-agent architecture. Finally, Table 1 provides a summary of recommendation works using MAS, categorized by their application domains, the techniques they use, and the objects they recommend.

In the aforementioned works, intelligent reconciliations are necessary for coordination among agents, as it allows them to communicate with each other

Table 1 Summary of recommendation works based on MAS

Works	Application Domains	Techniques Used	Recommended Objects
[15]	Tourism	Demographic based filtering and content-based filtering	Tourist locations, activity planning
[16]	Tourism	Reputation-based collaborative filtering	Hotels, locations, and restaurants
[17]	Web pages	Association rules and collaborative filtering	Next web page
[18]	Points of interest	Content-based filtering and sentiments analysis	Next location
[19]	Streaming platform	Content-based filtering	Movies
[20]	Streaming platform	Collaborative filtering	Movies and series
[21]	E-learning	Content-based filtering	Online courses,
[22]	E-commerce	Random forest, neural networks, support vector machines, k-nearest neighbors, and Naïve Bayes.	products/services
Our Model	**Tourism**	**Collaborative filtering**	**Points of interest**

while managing their conflicts [23]. However, there are several negotiation approaches, such as game theory, heuristics, argumentation, and voting. Game theory uses agents to maximize their gains. For this reason, each agent must anticipate the behavior of all others to find the optimal solution, which requires very high computational cost [24]. Heuristics have emerged to solve this problem, as each agent relies on its reasoning and strategies for decision-making without seeking optimal solution [25]. Argumentation is a method that allows agents to present proposals accompanied by detailed explanations, justifying their decisions to accept or reject these proposals. Therefore, each agent has the opportunity to accompany its proposals with arguments to explain why the others should consider them favorably. However, this technique requires very high communication costs [26]. Finally, voting is a technique used in collective decision theory or social choice theory that allows the selection of an alternative from among various possible options. It provides participants with the opportunity to express their preferences from a set of solutions. This technique appears to be well-suited to our reconciliation context, as the coordinating agent in our case must merge multiple lists of recommended POIs into a single global list [27].

Previous works highlight the importance of agent autonomy and their ability to reconcile each other in order to accomplish global tasks. However, no work in the literature addresses the cold-start problem in recommender systems caused by users' indifference towards evaluating POIs. To mitigate this issue, we explore in this paper the possibility of using MAS to exploit the temporal and geographic behavior of tourist and their social relationships with other users.

3 Proposed Model

To mitigate the cold-start problem in POIs recommendations related to user indifference and biased POI ratings, we propose in this section a MSR-PRA. This system comprises five intelligent, autonomous, and cooperative agents: (1) ContextAg agent, which collects the contextual information (time, location) of each user; (2) RatAg agent, which generates a fist list of recommended POIs using Pearson similarity; (3) FrAg agent, which provides a second list of recommended POIs based on Jaccard similarity; (4) TrAg agent, which derives a third list by leveraging trust relationships between users; and (5) coordinator agent (CorAg), which selects the most relevant POIs by merging these three lists of recommendations (see Figure 1).

3.1 ContextAg Agent

ContextAg agent collects the user's contextual data, such as location deduced via the Smartphone's GPS, date and time of day, and weather conditions, which can be retrieved using a weather API. This agent also instantly records all the user's activities through (1) ratings they assign to POIs, (2) check-ins they perform during their visit, and (3) declaration of trust degrees associated with each of their friends. Finally, the ContextAg agent structures the collected data into a rating matrix, a check-in matrix, and trust matrix (arrows A0, B0, and C0 in Figure 1).

3.2 RatAg Agent

RatAg agent uses collaborative filtering to suggest POIs for a given user to visit by leveraging their similarity with other users who have rated common POIs [28]. This agent calculates the similarity between two users, noted $SimP(u, v)$ using Pearson correlation [29], as indicated in Equation (1)

$$SimP(u,v) = \frac{\sum_{j \in Iu \cap Iv} (r_{uj} - \overline{r}_u) \cdot (r_{vj} - \overline{r}_v)}{\sqrt{\sum_{j \in Iu \cap Iv} (r_{uj} - \bar{r}_u)^2} \cdot \sqrt{\sum_{j \in Iu \cap Iv} (r_{vj} - \bar{r}_v)^2}} \tag{1}$$

where r_{uj} denotes the rating assigned by user u to POI j, r_{vj} denotes the rating given by user v to POI j, $\overline{r}_u$ and $\overline{r}_v$ are the average ratings of user u and v, respectively, computed over all the POIs they have rated. I_u and I_V denote the sets of POIs rated by users u and v, respectively, while $I_u \cap I_v$ represents the set of POIs rated by both users u and v.

After calculating the similarity between users (arrows A.1 and A.2 in Figure 1), RatAg agent proceeds to predict the ratings of POIs (arrow A.3 in Figure 1) using equation (2)

$$pred(u,p) = \bar{r}_u + \frac{\sum_{v \in Vu} (r_{vp} - \bar{r}_v) \cdot SimP(u,v)}{\sum_{v \in Vu} |SimP(u,v)|} \tag{2}$$

where $pred(u, p)$ denotes the predicted rating of user u for POI p, $\overline{r}_u$ is the average ratingof user u computed over all the POIs they have rated, V_u denotes the set of neighboring (similar) user of u, v is a neighboring user in V_u, r_{vp} is the rating assigned by user v to POI p, $\overline{r}_v$ is the average rating of user v, and $SimP(u, v)$ represents the Pearson similarity between users u and v, as defined in Equation (1). After calculating the user-user similarity matrix and prediction matrix, RatAg agent selects the list of top-k POIs (noted L_{UA})

and sends it to the coordination agent CorAg for a global recommendation (arrows A.4 and A.5 in Figure 1).

3.3 FrAg Agent

FrAg exploits the fact that user' social behavior is influenced by their friends [30]. For this reason, the friends of a given user can be a valuable source of recommendations. Therefore, this agent uses friendship relationships to provide recommendations, assuming that users who have the same friends are similar. This type of similarity, noted as $SimJ(u, v)$, is calculated using Equation (3)

$$SimJ(u, v) = \frac{|F_u \cap F_v|}{|F_u \cup F_v|} \tag{3}$$

where F_u and F_v denote the sets of friends of users u and v, respectively. $|Fu \cap Fv|$ is the number of mutual friends shared by user u and v, whereas $|F_u \cup F_v|$ represents the total number of distinct friends belonging to either user.

After calculating the similarity between users (arrows B.1 and B.2 in Figure 1), FrAg agent proceeds to predict the ratings of POIs (arrow B.3 in Figure 1) using Equation (4)

$$pred(u, p) = \bar{r}_u + \frac{\sum_{v \in Vu}(r_{vp} - \bar{r}_v) \cdot SimJ(u, v)}{\sum_{v \in Vu} |SimJ(u, v)|} \tag{4}$$

where $pred(u, p)$ denotes the predicted rating of user u for POI p, $\overline{r}_u$ is the average ratingof user u computed over all the POIs they have rated, V_u denotes the set of neighboring user of u, v is a neighboring user in V_u, r_{vp} is the rating assigned by user v to POI p, $\overline{r}_v$ is the average rating of user v, and $SimJ(u, v)$ represents the Jaccard similarity between users u and v, as defined in Equation (3). After calculating the user-user similarity matrix and prediction matrix, FrAg agent selects the list of top-k POIs (noted $\mathrm{L_{FA}}$) and sends it to the coordination agent CorAg for a global recommendation (arrows B.4 and B.5 in Figure 1).

3.4 TrAg Agent

TrAg agent generates recommendations based on the explicit trust expressed by each user towards other users. This type of trust, derived from friendship relationships between users, can be used to fill the values of the user-user similarity matrix. Using the values from this matrix (arrows C.1 and C.2 in

Figure 1), TrAg agent calculates the predicted ratings of POIs (arrow C.3 in Figure 1) using Equation (5)

$$pred(u,p) = \frac{\sum_{v \in Lut} r_{vp} \cdot d_{uv}}{\sum_{v \in Lut} d_{uv}} \tag{5}$$

where $pred(u,p)$ denotes the predicted rating of user u for POI p, L_{ut} denotes the set of users trusted by u, v is a trusted user belonging to L_{ut}, r_{vp} is the rating assigned by user v to POI p, and d_{uv} denotes the trust degree between users u and v, which serves as the weighting factor in the prediction model. After calculating the user-user similarity matrix and prediction matrix, the TrAg agent selects the list of top-k POIs (noted $\mathrm{L_{TA}}$) and sends it to the coordination agent CorAg for a global recommendation (arrows C.4 and C.5 in Figure 1).

3.5 Coordinator Agent

In our system, three agents (RatAg, FrAg, TrAg) utilize three types of similarity – Pearson similarity [29], Jaccard similarity [31], and trust-based similarity [32] – to generate three recommendation lists of POIs ($\mathrm{L_{UA}}$, $\mathrm{L_{FA}}$, and $\mathrm{L_{TA}}$). Consequently, the coordinator agent, CorAg, is responsible for merging and sorting these three lists, which contain POIs to recommend to the user. To achieve this, the CorAg agent employs the Borda voting system, allowing each agent to rank its alternatives in order of preference by assigning a positive score n to the first POI in the list, $n-1$ points to the second POI, and 0 points to the last POI (where n must be less than or equal to the number of alternatives) [33].

For example, suppose the CorAg agent receives three lists: $\mathrm{L_{UA}}$, $\mathrm{L_{FA}}$, and $\mathrm{L_{TA}}$, generated by the RatAg, FrAg, and TrAg agents. These lists contain the top@10 POIs to recommend, as shown in Table 2.

Table 2 illustrates how the Borda score method is used by the CorAg agent to generate the final list of POIs that reconciles the results obtained by the three agents RatAg, FrAg, and TrAg. Algorithm 1 formulates the reconciliation method used by CorAG agent to merge the lists $\mathrm{L_{UA}}$, $\mathrm{L_{FA}}$, and $\mathrm{L_{TA}}$ returned by each of the recommendation agents RatAg, FrAg, and TrAg, into a final merged list $\mathrm{L_F}$ based on the Borda technique.

This algorithm consists of the following steps:

1. Receive the three top-k POI lists returned by the agents RatAg, FrAg, and TrAg.
2. Assign a Borda score to all POIs in the three lists $\mathrm{L_{UA}}$, $\mathrm{L_{FA}}$, and $\mathrm{L_{TA}}$.

Table 2 Calculating Borda score

L_{UA}	L_{TA}	L_{FA}	N	L_F	Borda Score
POI_7	POI_{11}	POI_{18}	9	POI_{14}	$POI_{14} = 8+7+7 = 22$
POI_{14}	POI_{12}	POI_{11}	8	POI_{16}	$POI_{16} = 7+6+6 = 19$
POI_{16}	POI_{14}	POI_{14}	7	POI_{11}	$POI_{11} = 9+8 = 17$
POI_{13}	POI_{16}	POI_{16}	6	POI_{12}	$POI_{12} = 8+5+4 = 17$
POI_{18}	POI_6	POI_{12}	5	POI_{18}	$POI_{18} = 9+5 = 14$
POI_{12}	POI_{10}	POI_{13}	4	POI_{13}	$POI_{13} = 6+1+4 = 11$
POI_{20}	POI_{17}	POI_4	3	POI_7	$POI_7 = 9$
POI_9	POI_4	POI_{17}	2	POI_6	$POI_6 = 5+0 = 5$
POI_1	POI_{13}	POI_{10}	1	POI_{10}	$POI_{10} = 4+1 = 5$
POI_6	POI_9	POI_9	0	POI_{17}	$POI_{17} = 3+2 = 5$
POI_8	POI_{21}	POI_2	/	POI_4	$POI_4 = 3+2 = 5$
POI_{19}	POI_{15}	POI_8	/	POI_9	$POI_9 = 2+0+0 = 2$
POI_{23}	POI_2	POI_1	/	POI_1	$POI_1 = 1$

3. Calculate the sum of Borda scores for each POIs based on its presence in the three lists L_{UA}, L_{FA}, and L_{TA}.
4. Merge the three lists L_{UA}, L_{FA}, and L_{TA} into the final list L_F.
5. Sort the POIs in L_F based on their Borda scores.
6. Select the top-k POIs based on their Borda scores in L_F.
7. Send the final list L_F to the current user (U_C).

Algorithm 1 Reconciliation of RatAg, FrAg, and TrAg agents using the Borda method

Input: MR: Users/POIs rating matrix, **MA**: Users/Users relationship Matrix, **MT**: Users/Users Trust Matrix.
Output: L_{UA} POI list returned by RatAg agent, L_{FA} POI list returned by FrAg agent, L_{TA} POI list returned by agent TrAg, L_F the resulting list from merging lists L_{UA}, L_{TA} and L_{FA} returned by CorAg agent.
Points_per_list: Points matrix per POI and per the L_{UA}, L_{TA}, L_{FA} and L_F List
Var
N: integer // number of similar users,
K: integer // number of recommended POIs,
Uc: integer // current user index.
SB: integer // Borda score.
N: integer // maximum Borda score.
R: integer // item rank in the list
1: **Begin**
2: // three list calculation : L_{UA}, L_{FA} and L_{TA} for each current user
3: **Foreach** Uc in Users/POIs rating matrix:
4: L_{UA} = Result of RatAgAgent()

L_{FA} = Result of FrAgAgent()
L_{TA} = Result of TrAgAgent()
EndFor
// Calculating Borda scores for each POI of threelists: L_{UA}, L_{TA} and L_{FA}
Foreach L in lists L_{UA}, L_{TA} and L_{FA}:
Foreach POI in the list L:
    Points [L, POI] = K − R
EndFor
EndFor
// Calculation of the sum of POI scores from the lists L_{UA}, L_{TA} and L_{FA} into L_F list.
Foreach POI in the Points_per_listMatrix:
Foreach L in lists L_{UA}, L_{TA} and L_{FA}:
Points_per_list [L_F, POI] = Points_per_list [L_F, POI] + Points_per_list [L, POI]
EndFor
EndFor
// Sort POIs from the LF list based on their Borda scores
L_F = descending-sorting (L_F)
End

4 Experimentation and Results

In this section, we present the experimentation and results of our recommendation model. We begin with a description of the dataset. Next, we detail the metrics used for evaluation and identify the hyperparameters to be considered. Then, we outline the procedure to be followed for the evaluation. Finally, we analyze and discuss the results.

4.1 Dataset

We evaluate the proposed model on a dataset collected from several master's projects. During these projects, we have developed RS for POIs located in the city of Mostaganem, Algeria. The dataset consists of a set of users, a set of POIs, the ratings given by each user to the POIs, the list of user's friends, and the explicit trust degree expressed by a user toward their friends. Table 3 provides a description of our dataset.

4.2 Evaluation Metrics and Hyperparameters

In this section, we present the metrics that will be used to evaluate our model. Then, we detail the hyperparameters that will be considered in the experiments.

Table 3 Description of dataset

Column Name	Type/Value	Description
User_ID	String	Identifier of each user
POI_ID	String	Identifier of each POI
Rating	$\{1,2, 3, 4, 5\}$	Rating assigned by a user to a given POI
Friend	String	Identifier of the direct friend of a given user
Trust_Degree	$\{1, 2, 3, 4, 5\}$	Trust degree assigned by a given user to their friend

4.2.1 Evaluation metrics

To assess the effectiveness of the proposed model, we used three metrics: RMSE, Precision@K, and Recall@K, where K represents the number of recommended results.

4.2.2 RMSE

RMSE measures the difference between the actual ratings provided by users and the ratings predicted by the RS [34]. We use RMSE per query, noted $\mathrm{RMSE}_{\mathrm{request}}$, which calculates the RMSE for a given user per query. This metric is computed using Equation (6)

$$RMSE_{request} = \sqrt{\frac{\sum_{i=1}^{m}(r_i - \hat{r}_i)^2}{m}} \tag{6}$$

where r_i is the real rating i given by the user, $\hat{r}_i$ denotes the rating i predicted by RS, and m is the number of ratings provided by a given user per query.

4.2.3 Precision@K and Recall@K

Precision and Recall are commonly used metrics to evaluate RS based on a Top-K list of POIs [35]. Precision@K measures the proportion of relevant POIs among the top-K POIs returned by the system. This metric is computed using Equation (7)

$$Precision@K = \frac{1}{|U|} \sum_{u \in U} \frac{|Rec_u \cap Pert_u|}{|Rec_u|} \tag{7}$$

where Rec_u represents the set of POIs recommended to user u, $Pert_u$ is the set of relevant POIs for user u, and U denotes the set of users.

Recall@K represents the proportion of relevant items identified from the total set of relevant items available in the data, appearing among the top-K

Table 4 Evaluation hyperparameters of MSRPRA system

Symbol	Value	Description
K	5, 10, 15	Number of recommended POIs
N	1, 2, 3,...,15	Number of similar users
%Training Set	80%	Dataset for training model
%Test Set	20%	Dataset used on evaluation phase

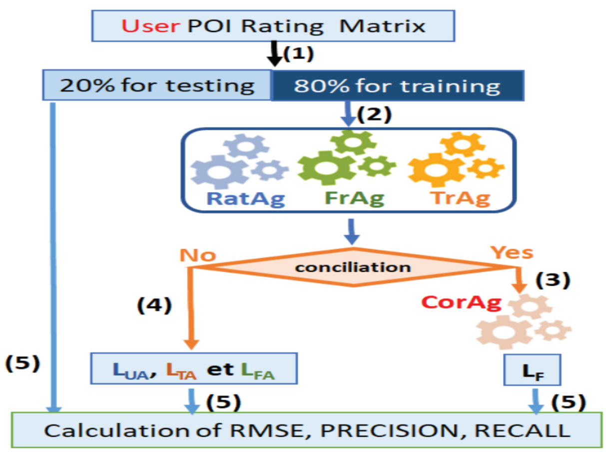

Figure 2 Experimental procedure for MSRPRA model.

results provided by the system. This metric is computed using Equation (8)

$$Recall@K = \frac{1}{|U|}\sum_{u\in U}\frac{|Rec_u \cap Pert_u|}{|Pert_u|} \tag{8}$$

4.2.4 Hyperparameters

To evaluate the performance of our model, we used the hyperparameters described in Table 4.

4.3 Experimental procedure

To evaluate our system, we randomly select 20% of each user's evaluated POIs as truth base for testing. The remaining portion (80%) for each user constitute a training dataset. Next, the model calculates the predicted rating of each unrated POI for each user and returns the Top@K (K = 5, 10, 15) POIs ranked by the predicted ratings. Finally, the RMSE, Precison@K, and Recall@K measures are used to evaluate the performance model. Figure 2 describes the main steps involved in our experimental approach for the MSRPRA model.

1. Divide the User-POI matrix into training and test sets.
2. Each agent (RatAg, FrAg, TrAg) calculates the similarities between the selected user and other users according to the method used. Then, selects the N most similar users to the selected user to predict the ratings.
3. After receiving the lists $\mathrm{L_{UA}}$, $\mathrm{L_{TA}}$, and $\mathrm{L_{FA}}$, the CorAg agent merges these three lists into the final list ($\mathrm{L_F}$) by applying the reconciliation principle of the Borda method.
4. The CorAg agent ignores the reconciliation principle and keeps the three lists $\mathrm{L_{UA}}$, $\mathrm{L_{TA}}$, and $\mathrm{L_{FA}}$.
5. Calculate Precison@K, Recall@K, RSME using $\mathrm{L_{UA}}$, $\mathrm{L_{TA}}$, $\mathrm{L_{FA}}$, $\mathrm{L_F}$ lists and the dataset test.

4.4 Results and Discussion

In this section, the results obtained by our MSRPRA system are compared with traditional similarity methods: Pearson similarity, noted SPearson, Jaccard similarity, noted SJaccard and, another explicit trust-based method, noted Trust. The purpose of this comparison is to study the contribution of agent reconciliation in the MSRPRA model for POI recommendation. For this reason, we have used reference metrics such as Recall@K, Precision@K, and RMSE to evaluate the results.

4.4.1 Comparison of the quality of MSRPRA model results

To compare the quality of POI recommendations, we calculated Recall@k and Precision@K for the methods (Pearson similarity, Jaccard similarity, Trust) as well as for the proposed MSRPRA system, as shown in Figures 3 and 4.

Figures 3 and 4 show that the MSRPRA model has a significant improvement compared to the SPearson, SJaccard, and Trust methods in terms of Precision@5, Recall@5, Precision@10, and Recall@10. However, we observe that the SPearson method shows a slight improvement over our model in terms of Precision@15 and Recall@15.

4.4.2 Comparison of the accuracy of MSRPRA model results

To evaluate the accuracy of POI recommendations, we compared our MSRPRA model with SPearson, SJaccard, and Trust methods in terms of RMSE, as shown in Figure 5.

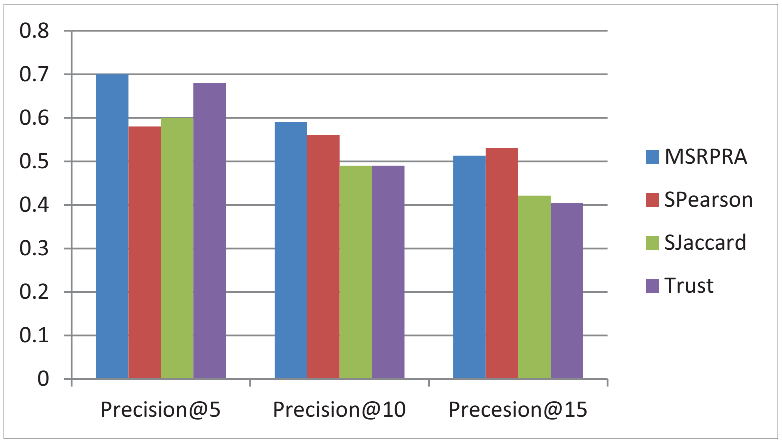

Figure 3 Comparison of SPearson, SJaccard, and Trust methods with MSRPRA using Precision parameter.

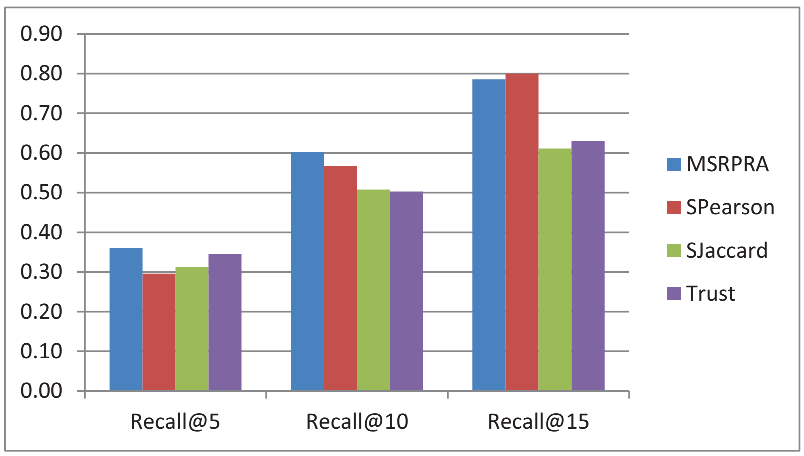

Figure 4 Comparison of SPearson, SJaccard, and Trust methods with MSRPRA using Recall parameter.

Figure 5 shows that our MSRPRA model provides an acceptable accuracy compared to the SPearson, SJaccard, and Trust methods in terms of RMSE.

4.4.3 Results analysis and discussion

The performance of our system is compared with methods that use Pearson similarity, Jaccard similarity, and the trust measure using Precision@K, Recall@K, and RMSE. The results of these comparisons are summarized in

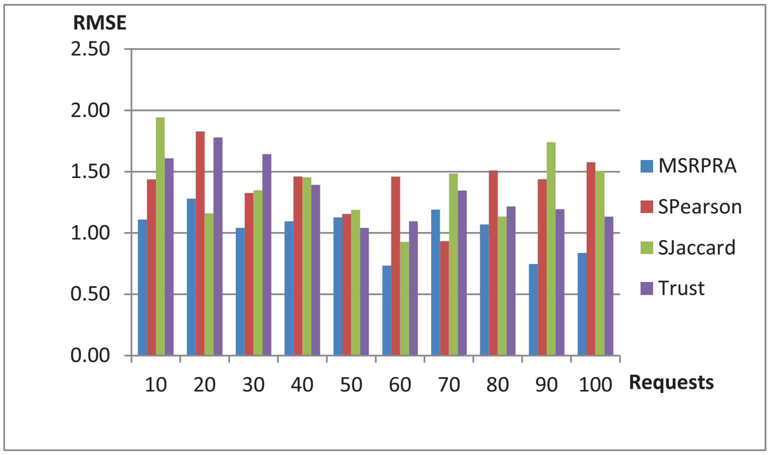

Figure 5 Comparison of SPearson, SJaccard, and Trust methods with MSRPRA.

Table 5 Comparison of MSRPRA model with SPearson, SJaccard, and Trust methods using Avg_Precision, Avg_Recall, and Avg_RMSE

	MSRPRA	SPearson	SJaccard	Trust
Avg_RMSE	1.179	1.337	1.28	1.278
Avg_Precision	0.594	0.567	0.544	0.549
Avg_Recall	0.577	0.557	0.524	0.527

Table 5, using the Precision average noted Avg_Precision, the Recall average, noted Avg_Recall, and the RMSE average, noted Avg_RMSE.

The results showed that MSRPRA outperforms the other methods. Specifically, we observed a 7.38% improvement in Precision and a 7.73% improvement in Recall. Similarly, for RMSE, our system demonstrates better recommendation accuracy compared to the Pearson similarity, trust measure, and Jaccard similarity methods. This indicates that the fusion performed by the CorAg agent successfully selected the most relevant POIs from the three lists provided by the RatAg, FrAg, and TrAg agents.

On the other hand, the use of the MSRPRA model helped mitigate the cold-start problem, as each agent utilizes its own data source and a recommendation algorithm different from those of the other agents.

5 Conclusion

In this paper, we designed a MSRPRA. Our system includes five agents (ContextAg, RatAg, FrAg, TrAg, CorAG) that use three recommendation algorithms. The first algorithm is based on Pearson similarity, the second algorithm used Jaccard similarity, and the third algorithm relies on trust relationships between users. These algorithms exploit three types of data: user ratings, friendship relationships, and explicitly declared trust degrees between users. Each agent returns a list of the most relevant POIs. The three lists ($\mathbf{L_{UA}}$, $\mathbf{L_{TA}}$, and $\mathbf{L_{FA}}$) are merged by the coordinator agent CorAg using the Borda voting technique into a final list ($\mathbf{L_F}$). The results of the experimental tests show that MSRPRA outperforms the algorithms using Pearson similarity, Jaccard similarity, and trust measure, in terms of RMSE, Precision, and Recall. This model also addresses the issue of data sparsity that hinders cold start in RS and causes user dissatisfaction. Finally, future research could focus on improving this system by integrating other algorithms such as Deep Learning [36] or incorporating additional types of data such as user reviews and check-ins.

References

[1] Y. Yu, X. Chen, "A survey of point-of-interest recommendation in location-based social networks," In Workshops at the Twenty-Ninth AAAI Conference on Artificial Intelligence, April 2015.

[2] J. D. Kelleher, *Deep Learning*. MIT Press, 2019.

[3] K. Thakaeng, P. Temdee, S. Wicha, "A decision tree-based advisory recommendation system for dental students," Journal of Mobile Multimedia, 21(01): 179–196, 2025. https://doi.org/10.13052/jmm1550-4646.2117.

[4] D. Gavalas, C. Konstantopoulos, K. Mastakas, G. Pantziou, "Mobile recommender systems in tourism," Journal of Network and Computer Applications, 39: 319–333, 2014.

[5] V. Jain, B. Malviya, S. Arya, "An overview of electronic commerce (e-Commerce)," Journal of Contemporary Issues in Business and Government, 27(3): 665–670, 2021.

[6] F. Ricci, L. Rokach, B. Shapira, *Recommender Systems Handbook*. Springer, 2015. https://doi.org/10.1007/978-1-4899-7637-6.

[7] Z. Abbasi-Moud, H. Vahdat-Nejad, J. Sadri, "Tourism recommendation system based on semantic clustering and sentiment analysis," Expert Systems with Applications, 167: 114324, 2021.

[8] J.B. Griesner, T. Abdessalem, H. Naacke, "Intégration des Influences Géographique et Temporelle pour la Recommandation de Points d'Intérêt," In Extraction et Gestion des Connaissances, 153–158, 2016.

[9] D. Lian, C. Zhao, X. Xie, G. Sun, E. Chen, Y. Rui. "Geomf: Joint geographical modeling and matrix factorization for point-of-interest recommendation," KDD'14, 2014.

[10] L. Cai, J. Xu, J. Liu, T. Pei, "Integrating spatial and temporal contexts into a factorization model for POI recommendation," International Journal of Geographical Information Science, 32(3): 524–546, 2018. DOI:10.1080/13658816.2017.1400550.

[11] Y. Liu, M. Yin, X. Zhou, "A collaborative filtering algorithm with intragroup divergence for POI group recommendation," Applied Sciences, 11(12): 5416, 2021.

[12] Z. Maghni Sandid, Z. Slama, N. Dennouni, "Toward mitigating tourists' indifference in POI recommendation systems using MAS," NCAIA'2023, 75, 2023.

[13] Z. F. Lansdowne, B. S. Woodward, "Applying the Borda ranking method," Air Force Journal of Logistics, 20(2): 27–29, 1996.

[14] A. Selmi, Z. Brahmi, M. Gammoud, "Multi-agent recommender system: State of the art," In Proceedings of the 16th International Conference on Information and Communications Security, 2014.

[15] L. Sebastia, A. Giret, I. Garcia, "A multi agent architecture for tourism recommendation," In Trends in Practical Applications of Agents and Multiagent Systems: 8th International Conference on Practical Applications of Agents and Multiagent Systems, 547–554, 2010.

[16] P. Bedi, S. K. Agarwal, V. Jindal, Richa, "MARST: Multi-Agent Recommender System for e-Tourism Using Reputation Based Collaborative Filtering," In Databases in Networked Information Systems: 9th International Workshop, DNIS 2014, Aizu-Wakamatsu, Japan, March 24–26, 2014.

[17] A. J. Morais, E. Oliveira, A. M. Jorge, "A multi-agent recommender system," In Distributed Computing and Artificial Intelligence: 9th International Conference," 281–288, Nov. 2012. Springer Berlin Heidelberg. doi:10.1007/978-3-642-28765-7_33.

[18] C. Cavallaro, G. Verga, E. Tramontana, O. Muscato, "Multi-agent architecture for point of interest detection and recommendation," In Proceedings of the CEUR Workshop Proceedings, 2404, 98–104, 2019.

[19] A. M. Mansour, M. A. Obeidat, B. Hawashin, "A novel multi agent recommender system for user interests extraction," *Cluster Computing*, 26(2): 1353–1362, 2023. doi:10.1007/s10586-022-03655-7.

[20] H. Kaur, H. Kaur, A. Singh, "Multi-agent based recommender system for Netflix," In Proceedings of International Conference on IoT Inclusive Life (ICIIL 2019), NITTTR Chandigarh, India, 211–221, 2020.
[21] M. Amane, K. Aissaoui, M. Berrada, "A multi-agent and content-based course recommender system for university e-learning platforms," In International Conference on Digital Technologies and Applications, 663–672, Jan. 2021. Cham: Springer International Publishing. doi:10.1007/978-3-030-73882-2_60.
[22] A. Alhejaili, S. Fatima, "Multi-agent recommender system," In Recent Advances in Agent-based Negotiation: Formal Models and Human Aspects. Springer Singapore, 103–119, 2021. doi:10.1007/978-981-16-0471-3_7.
[23] A. Dorri, S. S. Kanhere, R. Jurdak. "Multi-agent systems: A survey," IEEE Access, 6: 28573–28593, 2018.
[24] D. G. Pruitt, "Negotiation behavior," Academic Press, 2013.
[25] S. Kraus, "Negotiation and cooperation in multi-agent environments," Artificial Intelligence, 94(1–2): 79–97, 1997.
[26] A. R. Panisson, P. McBurney, R. H. Bordini, "A computational model of argumentation schemes for multi-agent systems," Argument & Computation, 12(3): 357–395, 2021.
[27] J. Pitt, L. Kamara, M. Sergot, A. Artikis, "Voting in multi-agent systems," The Computer Journal, 49(2): 156–170, 2006.
[28] H. Papadakis, A. Papagrigoriou, C. Panagiotakis, E. Kosmas, P. Fragopoulou, "Collaborative filtering recommender systems taxonomy," Knowledge and Information Systems, 64(1): 35–74, 2022.
[29] A. K. Hossain, Z. Tasnim, S. Hoque, M. A. Rahman, "A recommender system for adaptive examination preparation using Pearson correlation collaborative filtering," International Journal of Automation, Artificial Intelligence and Machine Learning, 2(1): 30–43, 2021.
[30] J. D. Zhang, C. Y. Chow, "Geosoca: Exploiting geographical, social and categorical correlations for point-of-interest recommendations," In Proceedings of the 38th International ACM SIGIR Conference on Research and Development in Information Retrieval, 443–452, 2015.
[31] S. Bag, S. K. Kumar, M. K. Tiwari, "An efficient recommendation generation using relevant Jaccard similarity," Information Sciences, 483: 53–64, 2019.
[32] S. Medjroud, N. Dennouni, M. Loukam, "Towards a systematic point-of-interest recommendations based on trust between users deduced

from their ratings and check-ins in a LBSN," International Journal of Computing and Digital Systems, 16(1): 189–203, 2024.

[33] P. Wang, Y. Fu, P. Liu, B. Zhu, F. Wang, D. Pamucar, "Evaluation of ecological governance in the Yellow River basin based on Uninorm combination weight and MULTIMOORA-Borda method," Expert Systems with Applications, 235: 121–227, 2024.

[34] M. Hadjhenni, N. Dennouni, Z. Slama, "Toward a systematic evaluation approach of point-of-interest recommendation algorithms of a novel smart tourism tool," International Journal of Computing and Digital Systems, 15(1): 653–670, 2024.

[35] H. J. C. Pampin, H. Jerbi, M. P. O'Mahony, "Evaluating the relative performance of collaborative filtering recommender systems," Journal of Universal Computer Science, 21(13): 1849–1868, 2015.

[36] P. Jing, "Digital media visual recommendation system based on artificial neural network machine learning," Journal of Mobile Multimedia, 21(05): 967–996, 2025. https://doi.org/10.13052/jmm1550-4646.2156.

Biographies

Zoulikha Maghni Sandid is a PhD student in the computer science department at Sidi Bel-Abbes University, Algeria. She graduated in 2007. She received her post graduation degree in computer science at the University of Science and Technology Mohamed Boudiaf USTO-MB Oran, in 2011. Her research interests include ubiquitous computing, POI recommendation, orchestration of activities and Artificial Intelligence. She is a member of the ISIBA team of EEDIS laboratory at the computer science Department of Djillali Liabes University of Sidi Bel Abbes.

Zohra Slama is an associate professor in computer science and ISIBA team leader of the EEDIS laboratory at the Djilali Liabes University of Sidi Bel Abbes. She received her PhD from Djillali Liabes University of Sidi Bel Abbes. Her major research area is paradigms of Artificial Intelligence, mobile learning, orchestration of activities, POI recommendation, mobile computing, and advanced databases.

Nassim Dennouni is an associate professor at the Higher School of Management at Tlemcen, Algeria, since October 2022. He is ICAR team leader of the LIA Laboratory of the Hassiba Benbouali University of Chlef and he is also a member of the ISIBA team of EEDIS laboratory of Computer Science Department of Djillali Liabes University of Sidi Bel Abbes. He obtained his PhD and university accreditation from Djillali Liabes University in2016 and 2018, respectively. His primary research domains are Artificial Intelligence, ubiquitous computing, mobile learning, orchestration of activities, and POI recommendation.