




A Multi-Agent Conciliation Framework for Mitigating the Cold-Start Problem in Point of Interest Recommendation Systems



Zoulikha Maghni Sandid^{1,2*}, Zohra Slama¹, Nassim Dennouni^{1,3}

¹ EEDIS Laboratory, Department of Computer Science, Djilali LIABES University, Sidi Bel Abbes 22000, Algeria

² Computer Science Department, Abdelhamid BEN BADIS University of Mostaganem, Mostaganem 27000, Algeria

³ Second Cycle Department, Higher School of Management, Tlemcen 13000, Algeria

Corresponding Author Email: maghnisandid.z@yahoo.com

Copyright: ©2026 The authors. This article is published by IETA and is licensed under the CC BY 4.0 license (<http://creativecommons.org/licenses/by/4.0/>).

<https://doi.org/10.18280/isi.310306>

ABSTRACT

Received: 28 June 2025

Revised: 1 October 2025

Accepted: 20 March 2026

Available online: 31 March 2026

Keywords:

Point of Interest recommendation, cold-start problem, multi-agent system, Borda method, Condorcet method collaborative filtering, trust-based recommendation

Cold-start problems in Point of Interest (POI) recommendation systems pose significant challenges due to a lack of user interaction with POIs, which limits the availability of sufficient data for accurate recommendations. To address this issue, this paper proposes a Multi-Agent Conciliation Model for POI Recommendation Algorithms (MCMAPRA), a novel framework that incorporates three distinct recommendation agents based on Pearson similarity, Jaccard similarity, and explicit trust relationships between users. These agents generate separate lists of recommended POIs, which are then merged by two conciliation agents—BordaAg and CondorcetAg—using the Borda and Condorcet voting methods, respectively. Experimental results, derived from a real-world dataset from Yelp, demonstrate that MCMAPRA outperforms traditional recommendation methods such as Pearson and Jaccard similarity in terms of RMSE, accuracy, and recall. Furthermore, MCMAPRA (Borda) achieves an improvement of approximately 1% in accuracy and a 7% improvement in recall compared to the Condorcet variant. These findings underscore the potential of multi-agent systems in effectively mitigating cold-start problems and enhancing the quality of POI recommendations, even in environments with sparse data. This study provides a new perspective on how multi-agent systems and voting-based conciliation can be used to improve recommendation system performance, offering valuable insights for future developments in the field of collaborative filtering and recommendation algorithms.

1. INTRODUCTION

Point of Interest (POI) Recommender Systems (RSs) are important for helping users to explore places that may be interesting to them [1], such as restaurants [2], tourist sites [3], commercial spaces or cultural events [4]. POI recommendation makes it possible to suggest places to users based on their preferences and behaviors. To recommend places, RSs use several recommendation techniques such as Content-Based Filtering (CBF) [5], Collaborative Filtering (CF) [6] and Hybrid Filtering (HF) [7].

CF techniques exploit ratings provided by users, which serve as the starting point for the recommendation process [8, 9]. However, these techniques suffer from the problem of data scarcity caused by user indifference and biased evaluations, as the majority of users do not give great importance to the ratings of places already visited [10]. One way to solving this problem is to combine several recommendation algorithms running in parallel, exploiting other types of data. To achieve this goal, we have designed three recommendation agents, each using three different similarities-based algorithms and three different types of data. However, the conciliation of the POI lists obtained by these three recommendation agents poses a real challenge. To overcome this difficulty, the use of Multi-

Agent Systems (MAS) [11] makes it possible to combine the recommendation algorithms to obtain intelligent conciliation.

Several works in the literature have integrated multi-agent systems into recommendation systems to make them more intelligent. These systems, called Multi-Agent Recommendation Systems (MARS), are SRs integrating autonomous and cooperative agents to improve the quality of recommendations [12]. The authors Sebastia et al. propose MARS that recommends tourist locations in city of Valencia (Spain) using an activity plan. However, this system requires the intervention of the tourist at each stage to specify his needs and rate the places he has visited [13]. In the same context, authors Bedi et al. have proposed a MARS for e-tourism using reputation-based collaborative filtering, which allows services to be recommended, namely hotels, places and restaurants. The system assigns an agent to each service, and each agent returns a list of that service. However, although this processing exploits the strength of SMAs in the distribution and cooperation of recommendation tasks, this solution suffers from the sparse matrix problem [14]. To address this type of problem, a centralized MAS architecture was proposed in the study [15] to recommend locations to users. The server agent collects data from client agents to compute predictions of ratings. Then, the list of most relevant locations is sent to the

client agent, which in turn selects those with on a location perimeter. On the other hand, a video SR based on MASs has been proposed in the study [16]. Their system consists of seven agents (location, age, financial, identity, personality, needs and social) and an Information Center Agent (ICA) that collects and processes the information. Each agent specifies whether the POI is interesting or not, then the ICA ranks the POIs. However, in this system, each agent does not classify the films, but only expresses his or her interest or not over the film. In the same context, and in order to build user loyalty to Netflix platform, the authors Kaur et al. proposed a framework in the form of an MARS that suggests Ads for shows and films corresponding to user preferences. The system uses the K-Nearest Neighbors (KNN) supervised learning algorithm with cosine similarity. User ratings of films were used to calculate predictions. However, this type of platform suffers from the problem of data scarcity [17]. To support learners, MARS are used as e-learning platforms that are increasingly in use by students as the number of online courses increases day by day. For this reason, the authors Amane et al. have proposed a content-based SR using two agents. Their system works in two stages (similarity computation and reclassification by filtering). These two agents extract negative and positive comments in order to reclassify the list of top-k courses [18]. To increase the accuracy of the recommendation, the authors Alhejaili et al. [18] used an MARS consisting of six agents, five of which are each based on a machine learning method. The methods used are: Random forests, neural networks, support vector machines, KNN and naïve Bayes. The Contrat-Net protocol was used to manage interaction and negotiation between agents. Previous work emphasizes the importance of agents' autonomy and their ability to conciliate with each other to accomplish global tasks. However no work in the literature addresses the problem of cold start SRs caused by user indifference to POI evaluation.

For these reasons, we have proposed in this paper a Model of Conciliation Multi-Agent of POI Recommendation Algorithms (MCMAPRA). This model includes five agents: (1) three recommendation agents (Rec1Ag, Rec2Ag and Rec3Ag), each of them returns an ordered list of POIs, and (2) two conciliations agents named BordaAg and CondorcetAg, which merge these three lists and return a global list of the most relevant POIs using the voting technique [19] Borda [20] and Condorcet [21] respectively.

To evaluate the performance of our MCMAPRA model, we carried out experiments on a real dataset from Yelp dataset. The results show that the proposed model outperforms conventional methods in terms of RMSE, Precision and Recall. These methods rely solely Pearson similarity, Jaccard similarity and the trust measure.

The remainder of the paper is structured as follows: Section 2 describes our recommendation model; Section 3 details the experimentation phases, including an analysis and discussion of the results; finally, the conclusion provides a summary of our contributions and future perspectives.

2. PROPOSED MODEL

Cold start is a major issue in recommender systems. To mitigate this problem, which is linked to user indifference and biased POI ratings, we propose in this section a Model of Conciliation Multi-Agent of POI Recommendation Algorithms (MCMAPRA). This model consists of five intelligent, autonomous and cooperative agents: (1) the

Rec1Ag agent, which uses Pearson similarity [22] to suggest the first list of recommended POIs; (2) the Rec2Ag agent, which relies on Jaccard similarity to provide the second list; (3) the Rec3Ag agent, which produces the third list based on the trust level between users; (4) the BordaAg agent, the first conciliation agent, which merges the three POIs lists into a final list of the most relevant POIs using the Borda method; and (5) the CondorcetAg agent, the second conciliation agent, which uses the Condorcet method for the fusion (see Figure 1).

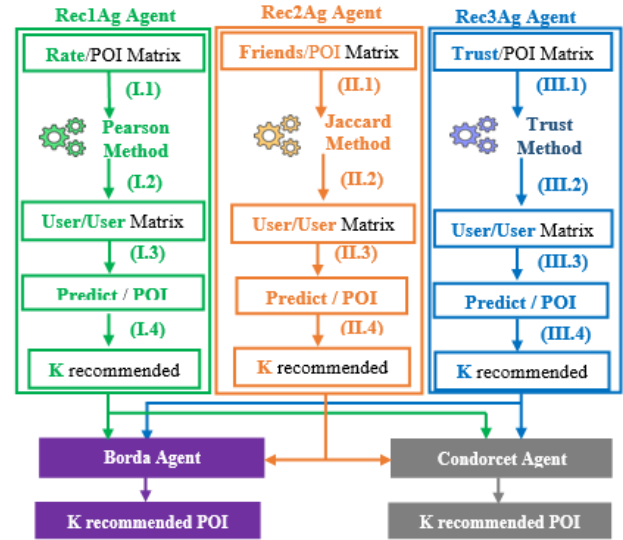


Figure 1. Functional architecture of our model

2.1 Rec1Ag agent

The agent named “Rec1Ag” is based on collaborative filtering to recommend POIs to a given user, relying on similar users who have rated common POIs [23]. To generate the list of POIs, the Rec1Ag agent proceeds as follows:

(1) Calculate the similarity between two users a and b (see arrows I.1 and I.2 in Figure 1), denoted $SP(a, b)$, using Pearson correlation [24], as shown in Eq. (1):

$$SP(a, b) = \frac{\sum_{j \in P_a \cap P_b} (rat_{aj} - \bar{rat}_a) \cdot (rat_{bj} - \bar{rat}_b)}{\sqrt{\sum_{j \in P_a \cap P_b} (rat_{aj} - \bar{rat}_a)^2} \cdot \sqrt{\sum_{j \in P_a \cap P_b} (rat_{bj} - \bar{rat}_b)^2}} \quad (1)$$

where:

- rat_{aj} : Rating given by user a to POI j .
- \bar{rat}_a : Average ratings of user a .
- $P_a \cap P_b$: The set of POIs rated by both a and b .

(2) Predict the ratings that the user might assign to unrated POIs (see arrow I.3 in Figure 1) using Eq. (2):

$$Pr(a, i) = \bar{rat}_a + \frac{\sum_{b \in U_a} (rat_{bi} - \bar{rat}_b) \cdot SP(a, b)}{\sum_{b \in U_a} |SP(a, b)|} \quad (2)$$

where:

- $Pr(a, i)$: Predicted rating of user a for POI i .
- U_a : The set of users similar to user a .
- rat_{bi} : Rating given by a similar user b to POI i .

(3) Select the top-k POI list (denoted P_{R1}) after calculating the similarity matrix and prediction matrix.

(4) Send the P_{R1} list to the conciliation agents “BordaAg” and “CondorcetAg” to generate a global recommendation.

2.2 Rec2Ag agent

The agent named “Rec2Ag” is based on the idea that users’ social behavior is influenced by their friendships [25]. Indeed, a given user’s friends can be a relevant source of recommendations. Therefore, the agent leverages friendship links to identify similar users, assuming that those who share the same friends have close preferences. To provide the list of POIs, the Rec2Ag agent proceeds as follows:

(1) Calculate the similarity between two users a and b (see the arrows II.1 and II.2 in the Figure 1), denoted $SJ(a,b)$, using Jaccard similarity [26], as shown in Eq. (3) below:

$$SJ(a, b) = \frac{|L_{f_a} \cap L_{f_b}|}{|L_{f_a} \cup L_{f_b}|} \quad (3)$$

where:

- L_{f_a} : User’s friends list a .
- L_{f_b} : User’s friends list b .

(2) Predict the ratings that the user might assign to unrated POIs (see the arrow II.3 in the Figure 1) using Eq. (4) below:

$$Pr(a, i) = \overline{rat_a} - \frac{\sum_{b \in U_a} (rat_{bi} - \overline{rat_b}) \cdot SJ(a,b)}{\sum_{b \in U_a} |SJ(a,b)|} \quad (4)$$

(3) Select the top- k POIs list (denoted P_{R2}) after calculating the similarity matrix and prediction matrix.

(4) Forward the P_{R2} list to the conciliation agents “BordaAg” and “CondorcetAg” to generate a global recommendation.

2.3 Rec3Ag agent

The agent named “Rec3Ag” generates recommendations based on the explicit trust that a user expresses towards other users [27]. This trust, inferred from friendship relationships, can be used to fulfill the user-user similarity matrix. To provide the list of POIs, the Rec3Ag agent proceeds as follows:

(1) Predict the ratings that the user a might assign to POIs that have not yet been rated (see the arrow III.3 in the Figure 1) based on the similarity matrix (see arrows III.1 and III.2 in the Figure 1), using Eq. (5) below:

$$Pr(a, i) = \frac{\sum_{b \in T_a} (rat_{bi}) \cdot dg_{ab}}{\sum_{b \in T_a} dg_{ab}} \quad (5)$$

where:

- T_a : The list of user a ’s friends whom they trust.
- dg_{ab} : Trust degree between user a and b .

(2) Select the top- k POIs list (denoted P_{R3}) after calculating the similarity matrix and prediction matrix.

(3) Send P_{R3} to the conciliation agents “BordaAg” and “CondorcetAg” to generate a global recommendation.

2.4 BordaAg agent

In our model, the Rec1Ag, Rec2Ag and Rec3Ag agents provide three lists of recommended POIs (P_{R1} , P_{R2} and P_{R3}). The role of the BordaAg agent is to provide a final “PF” list that merges the three lists. For this reason, the BordaAg agent uses the Borda method, which is a weighted voting method. It consists of raking the POIs in order of preferences, assigning the first POI in the list a positive number n of points, the second POI in the list “ $n-1$ ” points and the last POI in the list “0” point (the number n must be less than or equal to the number of POIs) [28].

For example, suppose that the agents Rec1Ag, Rec2Ag and Rec3Ag have sent their respective top@5 recommended POIs lists (P_{R1} , P_{R2} and P_{R3}) to the BordaAg agent. This information is shown in the following Table 1:

Table 1. Example of Borda Score Calculation (BSC)

P_{R1}	P_{R2}	P_{R3}	N	PF	BSC
POI ₂₁	POI ₁	POI ₇	4	POI ₁₀	POI ₁ =4+3=7
POI ₁	POI ₁₀	POI ₁₀	3	POI ₁	POI ₃ =1+2=3
POI ₁₀	POI ₇	POI ₃	2	POI ₇	POI ₄ =1=1
POI ₃	POI ₄	POI ₁₅	1	POI ₂₁	POI ₇ =4+2=6
POI ₁₅	POI ₂₁	POI ₁	0	POI ₃	POI ₁₀ =3+3+2=8
POI ₁₁	POI ₁₅	POI ₂₅	/	/	POI ₁₅ =0+0+1=1
POI ₂	POI ₃	POI ₁₂	/	/	POI ₂₁ =4+0=4

Table 1 above explains how the BordaAg agent calculates the score of each POI from the lists (P_{R1} , P_{R2} and P_{R3}) in order to merge them into a global recommendation (PF). The following Algorithm 1 outlines the method used by the BordaAg agent:

Algorithm 1: Conciliation using the Borda method

Input:

RM: Users/POIs rating matrix,
FM: Users/Users friendship Matrix,
TM: Users/Users Trust Matrix;

Output:

P_{R1} : POI list returned by Rec1Ag agent,
 P_{R2} : POI list returned by Rec2Ag agent,
 P_{R3} : POI list returned by agent Rec3Ag,

PF: list merging from P_{R1} , P_{R2} and P_{R3} returned by BordaAg agent,

Points_per_list: Points matrix per POI and per the P_{R1} , P_{R2} and P_{R3} and PF List;

Var

K: integer // number of recommended POIs,

Uc: integer // current user index.

R: integer // item rank in the list

Begin

// P_{R1} , P_{R2} and P_{R3} calculation: for each current user.

For each Uc in RM:

P_{R1} = Result of Rec1Ag_Agent()

P_{R2} = Result of Rec2Ag_Agent()

P_{R3} = Result of Rec3Ag_Agent()

End For

//calculating Borda scores for each POI of: P_{R1} , P_{R2} and P_{R3}

For each L in lists P_{R1} , P_{R2} and P_{R3} :

For each POI in the list L:

Points_per_list [L, POI] = K – R

End For

End For

// Calculation of the sum of the POI scores from the lists P_{R1} , P_{R2} and P_{R3} into PF list.

For each POI in Points_per_list Matrix:

For each L in lists P_{R1} , P_{R2} and P_{R3} :

Points_per_list [PF, POI] = Points_per_list [PF, POI] + Points_per_list [L, POI]

End For

End For

// Sort POIs from the PF list based on their Borda scores

PF = descending_sorting(PF)

End

2.5 CondorcetAg agent

The agent named ‘‘CondorcetAg’’, like the BordaAg agent, merge the P_{R1} , P_{R2} and P_{R3} lists received from Rec1Ag, Rec2Ag and Rec3Ag agents into a global recommendation list. The CondorcetAg agent uses a different voting method, namely the Condorcet method, which is a raking-based voting method. It involves dual comparisons of POIs, and the POI that surpasses all others is selected as the best POI [29].

Let’s take the following example: we assume that CondorcetAg agent receives the lists P_{R1} , P_{R2} and P_{R3} , from the three agents Rec1Ag, Rec2Ag and Rec3Ag, each containing the top@5 POIs to recommend, as shown in the following Table 2.

Table 2. Example illustrating the Condorcet ranking principle

P_{R1}	P_{R2}	P_{R3}
POI ₂₁	POI ₁	POI ₇
POI ₁	POI ₁₀	POI ₁₀
POI ₁₀	POI ₇	POI ₃
POI ₃	POI ₄	POI ₁₅
POI ₁₅	POI ₂₁	POI ₁

The POIs ranking according to the Condorcet principle involves the following steps:

- Calculate the dual matrix: (see Table 3)
- Calculate the victory scores and generate the final list: (see Table 4)

Tables 3 and 4 explain how the Condorcet ranking of POI is used by the CondorcetAg agent to obtain the final list. In what follows, Algorithm 2 outlines the conciliation method used by CondorcetAg to merge the three lists P_{R1} , P_{R2} and P_{R3} into the final list PF.

Algorithm 2: Conciliation using Condorcet method

Input:

RM: Users/POIS rating matrix,
 FM: Users/Users friendship matrix,
 MT: Users/Users trust matrix,
 L_{POI}: POIs list.

Output:

P_{R1} : POI list returned by Rec1Ag agent,
 P_{R2} : POI list returned by Rec2Ag agent,
 P_{R3} : POI list returned by agent Rec3Ag,
 PF: the resulting list.

Var

K: integer//number of POIs to recommender

Uc: integer // current user index.

DM: duels matrix of dimension m, m

Score: Dictionary // victory scores dictionary for each

POI.

Begin

// calculate P_{R1} , P_{R2} and P_{R3} lists for each current user.

For each Uc in RM matrix:

P_{R1} = Result of Rec1Ag_Agent ()

P_{R2} = Result of Rec2Ag_Agent ()

P_{R3} = Result of Rec3Ag_Agent ()

End For

DM[m] [m] = 0 //initialize the duels matrix

// calculate the duels matrix between POIs of P_{R1} , P_{R2} and P_{R3} .

For each L in P_{R1} , P_{R2} and P_{R3} list:

For each POI_i in list L:

For each POI_j ranked after POI_i in list L:

DM[POI_i][POI_j] += 1

End For

End For

End For

//calculate the victory scores for each POIs of P_{R1} , P_{R2} and P_{R3} lists in the PF list

Score[L_{POI}] = 0

For each POI_i in DM matrix:

For each POI_j in DM matrix:

IF DM [POI_i][POI_j] > DM[POI_j][POI_i] then:

Score [POI_i] += 1

End IF

End For

End For

//Sort the POIs in PF list based on their victory scores.

PF = descending_sort(L_{POI}, Score[L_{POI}])

End

Table 3. Example of dual matrix calculation

POIs	POI ₁	POI ₃	POI ₄	POI ₇	POI ₁₀	POI ₁₅	POI ₂₁
POI ₁	-	1	1	1	2	1	1
POI ₃	1	-	0	0	0	2	0
POI ₄	0	0	-	0	0	0	1
POI ₇	1	1	1	-	1	1	1
POI ₁₀	1	2	1	1	-	2	1
POI ₁₅	1	0	0	0	0	-	0
POI ₂₁	1	1	0	0	1	1	-

Table 4. Example of victory score calculation

POIs List	Victory Score	PF
POI ₁	Score=0+1+0+1+0+0= 2	POI ₇
POI ₃	Score=0+0+0+0+1+0=1	POI ₁₀
POI ₄	Score=0+0+0+0+0+1=1	POI ₁
POI ₇	Score=0+1+1+0+1+1=4	POI ₂₁
POI ₁₀	Score=0+1+1+0+1+0=3	POI ₄
POI ₁₅	Score=0+0+0+0+0+0=0	POI ₃
POI ₂₁	Score=0+1+0+0+1+0=2	POI ₁₅

3. EXPERIMENT AND RESULTS

In this section, we present the experimentation and results of our recommendation model. We begin with a description of the dataset. Then, we detail the evaluation procedure. Finally, we present the results and the discussion.

3.1 The dataset

For the evaluation of our model, we used the public Yelp dataset instead of our own dataset initially collected during the test phase of our prototype Point of Interest (POI) recommendation system in the city of Mostaganem, Algeria. The Yelp dataset contains information similar to that in our dataset, including a set of users, POIs (restaurants, entertainment venues, etc.), ratings assigned by users to POIs, as well as friendship relationships between users enabling an implicit or explicit degree of trust to be estimated. The collected data relates to the state of Nevada (NV) for the period 2005 to 2022. Table 5 gives a detailed description of the main features of the Yelp dataset used in our experiments.

Table 5. Description of the columns in our dataset

Column Names	Type/Values	Description
User_ID	String	The identifier of each user.
POI_ID	String	The identifier of each POI.
Rating	{1, 2, 3, 4, 5}	The rating assigned by a user to a give POI.
Friend	String	The identifier of the direct friend of a given user.
Trust_degree	{1, 2, 3, 4, 5}	The trust degree that a given user assigns to their friend.

3.2 Experimental procedure

To evaluate our recommendation model, we divide our dataset into a training set and a test set. We use 80% of dataset as the training set and the 20% remaining as the ground truth for test. The model then predicts the ratings for unrated POIs and returns the Top@K (K = 5, 10, 15) POIs. Finally, we evaluate the model's performance using the following metrics: RMSE [30], Precision@K [31] and Recall@K [32]. The main steps of our experimental procedure of the MCMAPRA model are described below:

- Split the Users/POIs matrix into a training set and a test set.
- Each agent (Rec1Ag, Rec2Ag and Rec3Ag) calculates the similarities, according to its respective method, between the current user and other users. It then selects the top N (1, 2, ..., 20) most similar users to predict ratings.
- After receiving the P_{R1} , P_{R2} and P_{R3} lists, the BordaAg agent merges them into the global list (PF) using the conciliation principle of Borda method.
- Similarly, the CondorectAg conciliation agent merges the three lists (P_{R1} , P_{R2} and P_{R3}) into the global list (PF) according to the principle of the Condorcet method.
- The BordaAg and CondorectAg agent ignore the conciliation principle and retain the three lists P_{R1} , P_{R2} and P_{R3} .
- Calculate Precision@K, Racall@K and RSME based on the P_{R1} , P_{R2} and P_{R3} lists, the PF list, and the test dataset.

3.3 Results and discussion

To assess the contribution of agent conciliation in our MCMAPRA model for POIs recommendation, we compared the results obtained by our model using the Borda and Condorcet methods separately with classical similarity methods: Pearson similarity, denoted SimPear, and Jaccard similarity, denoted SimJacc, as well as another explicit trust-based method, denoted ExpTrust. Evaluation was carried out using standard metrics such us Recall@K, Precision@K and RMSE.

3.3.1 Comparison of the result quality of the MCMAPRA model

To evaluate the quality of POI recommendations, we calculated Recall@K and Precision@K for the methods (Pearson similarity, Jaccard similarity, and Trust), as well as for the two MCMAPRA variants, MCMAPRA (Borda) and MCMAPRA (Condorcet) as shown in Figures 2 and 3.

Figure 2 and Figure 3 show that the MCMAPRA model outperforms the SimPear, SimJacc and ExptTrust methods in

terms of Precision, Recall. We can observe that MCMAPRA (Borda) provides better results than MCMAPRA (Condorcet) in terms of both Precision and Recall.

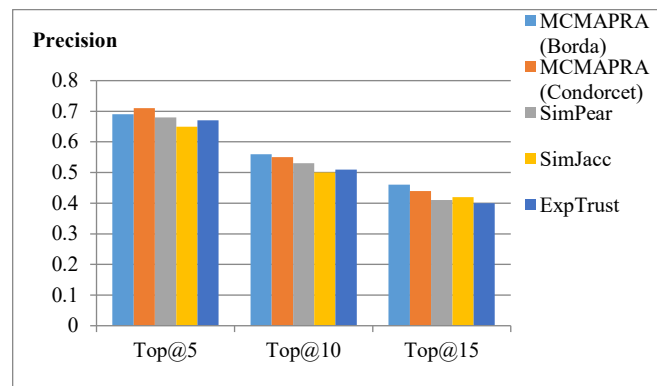


Figure 2. Comparison of SimPear, SimJacc and ExpTrust methods with MCMAPRA (Borda) and MCMAPRA (Condorcet) using the precision metric

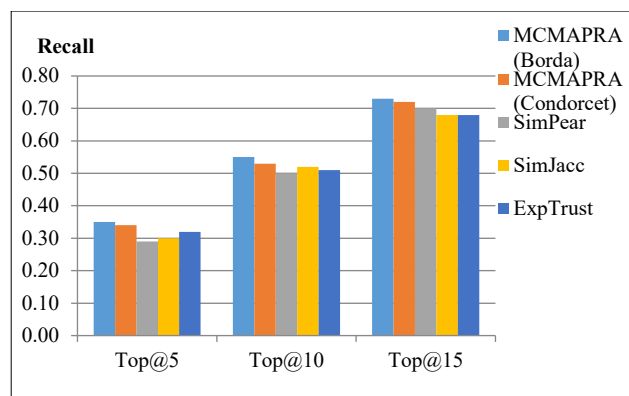


Figure 3. Comparison of SimPear, SimJacc and ExpTrust method with MCMAPRA (Borda) and MCMAPRA (Condorcet) using recall metric

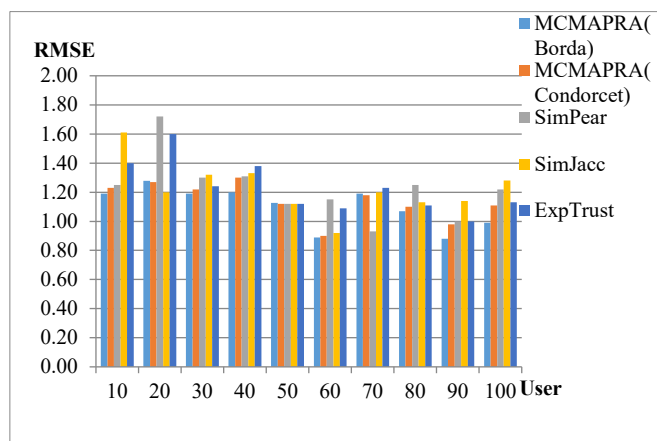


Figure 4. Comparison of SimPear, SimJacc and ExpTrust with MCMAPRA (Borda) and MCMAPRA (Condorcet)

3.3.2 Comparison of the accuracy of results MCMAPRA model

To evaluate the accuracy of POIs recommendations, we compared our MCMAPRA model (Borda and Condorcet) with the SimPear, SimJacc and ExpTrust methods in terms of RMSE, as shown in Figure 4.

Figure 4 shows that our MCMAPRA model provides

acceptable accuracy compared to SimPear, SimJacc and ExpTrust methods in terms of RMSE. When comparing the two variants, we observe that the MCMAPRA (Borda) achieves better results than MCMAPRA (Condorcet) in terms of RMSE.

3.3.3 Analysis and discussion of the results

The results of two variants of our model are compared with those of the methods using Pearson similarity, Jaccard similarity and the trust measure using Precision@K, Recall@K and RMSE. A summary of the results from these comparisons is provided in Table 6 below, using the average Precision (AvgPrec), the average Recall (AvgRec) and the average RMSE (AvgRMSE).

The results showed that both variants (MCMAPRA (Borda)

and (Condorcet)) outperformed the other methods. Specifically, we observed that MCMAPRA (Borda) achieved a 6.39% improvement in Precision and a 5.5% improvement in Recall, while MCMAPRA (Condorcet) showed a 6.95% improvement in Precision and a 7.29% improvement in Recall. Similarly, regarding RMSE, both variants demonstrated better recommendation accuracy than the Pearson similarity, trust-based, and Jaccard similarity methods. This indicates that the fusion carried out by the CondorcetAg or BordaAg agents successfully selected the most relevant POIs from the three lists provided by agents Rec1Ag, Rec2Ag and Rec3Ag. Additionally, MCMAPRA (Borda) achieved a 1% improvement in Precision and a 7% improvement in Recall compared to the MCMAPRA (Condorcet) model.

Table 6. Comparison of the MCMAPRA model variants with SimPear, SimJacc and ExpTrust methods using AvgPrec, Avgrec and AvgRMSE

	MCMAPRA (Borda)	MCMAPRA (Condorcet)	SimPear	SimJacc	ExpTrust
AvgRMSE	1.135	1.141	1.225	1.215	1.230
AvgPrec	0.570	0.567	0.541	0.531	0.527
AvgRec	0.540	0.530	0.500	0.506	0.503

This performance gap can be explained by the way each voting method exploits the recommendation lists. The Borda method assigns a score to every POI across all lists, which enables it to capture partial rankings and integrate more nuanced preference information from the three recommenders. As a result, it tends to highlight POIs that appear consistently across all algorithms, thus reflecting a stronger overall consensus. In contrast, the Condorcet method relies on pairwise comparisons, favoring POIs that win the most direct confrontations. However, this approach may lead to ties or cycles and can overlook relevant POIs when the recommenders disagree, particularly in contexts where lists are short or heterogeneous. Given that the most relevant POIs in our experiments were those jointly identified by all algorithms, the Borda method was better positioned to promote them, which explains its higher Precision (+1%) and Recall (+7%) compared to the Condorcet variant.

4. CONCLUSION

In this paper, we designed a Model of Conciliation Multi-Agent for POIs Recommendation Algorithms (MCMAPRA). Our model includes five agents: (1) three recommendations agents (Rec1Ag, Rec2Ag and Rec3Ag) and (2) two conciliations agents (CondorcetAg and BordaAg). The recommendation agents use three different recommendation algorithms based on three similarity methods: the first uses Pearson similarity, the second uses Jaccard similarity, and the third uses trust relationship between users as a similarity measure. These agents respectively leverage three types of data: user ratings, friendship relations, and explicitly declared trust degrees between users. The two conciliations agents merge the lists of the most relevant POIs (P_{R1} , P_{R2} and P_{R3}) provided by the recommendations agents using two voting methods: Condorcet and Borda, to produce a global list PF. Experimental results show that both MCMAPRA variants (Borda and Condorcet) outperform algorithms based on Pearson similarity, Jaccard similarity, and trust based similarity in terms of RMSE, Precision, and Recall. The MCMAPRA model addresses the cold-start problem of

recommender systems caused by data sparsity, which reduces recommendation quality.

Although our model significantly improves the overall quality of recommendations, it also has several limitations. First, while MCMAPRA improves performance during cold starts, its effectiveness may decrease in environments where data is extremely scarce, and where even consensus merging does not provide sufficient information to the recommendation agents. Second, the model introduces additional computational complexity due to the parallel execution of multiple recommendation systems and the conciliation process—particularly for the Condorcet method—which may limit its scalability for very large datasets. Third, the model encounters difficulties in real-time application, as multi-agent coordination and vote aggregation require additional processing that can hinder response speed. For future development, it would be worthwhile to explore additional data sources such as user reviews and POI check-ins, as well as to investigate adaptive mechanisms capable of dynamically selecting the most appropriate conciliation strategy based on the context [33].

REFERENCES

- [1] Wang, Z., Höpken, W., Jannach, D. (2025). A survey on point-of-interest recommendations leveraging heterogeneous data. *Information Technology & Tourism*, 27(1): 29-73. <https://doi.org/10.1007/s40558-024-00301-3>
- [2] Fakhri, A.A., Baizal, Z.K.A., Setiawan, E.B. (2019). Restaurant recommender system using user-based collaborative filtering approach: A case study at Bandung Raya Region. *Journal of Physics: Conference Series*, 1192(1): 012023. <https://doi.org/10.1088/1742-6596/1192/1/012023>
- [3] Abbasi-Moud, Z., Vahdat-Nejad, H., Sadri, J. (2021). Tourism recommendation system based on semantic clustering and sentiment analysis. *Expert Systems with Applications*, 167: 114324. <https://doi.org/10.1016/j.eswa.2020.114324>

- [4] Halimeh, H., Freese, F., Müller, O. (2023). Event recommendations through the lens of vision and language foundation models. In *RecTour@ RecSys*, pp. 39-60.
- [5] Son, J., Kim, S.B. (2017). Content-based filtering for recommendation systems using multiattribute networks. *Expert Systems with Applications*, 89: 404-412. <https://doi.org/10.1016/j.eswa.2017.08.008>
- [6] Valdiviezo-Diaz, P., Ortega, F., Cobos, E., Lara-Cabrera, R. (2019). A collaborative filtering approach based on Naïve Bayes classifier. *IEEE Access*, 7: 108581-108592. <https://doi.org/10.1109/ACCESS.2019.2933048>
- [7] Darban, Z.Z., Valipour, M.H. (2022). GHRS: Graph-based hybrid recommendation system with application to movie recommendation. *Expert Systems with Applications*, 200(6): 116850. <https://doi.org/10.1016/j.eswa.2022.116850>
- [8] Papadakis, H., Papagrigoriou, A., Panagiotakis, C., Kosmas, E., Fragopoulou, P. (2022). Collaborative filtering recommender systems taxonomy. *Knowledge and Information Systems*, 64(1): 35-74. <https://doi.org/10.1007/s10115-021-01628-7>
- [9] Cai, L., Xu, J., Liu, J., Pei, T. (2018). Integrating spatial and temporal contexts into a factorization model for POI recommendation. *International Journal of Geographical Information Science*, 32(3): 524-546. <https://doi.org/10.1080/13658816.2017.1400550>
- [10] Maghni Sandid, Z., Slama, Z., Dennouni, N. (2023). Toward Mitigating Tourists' Indifference in POI recommendation systems using MAS. In *National Conference on Artificial Intelligence and Its Applications, Algeria*, pp.75-79.
- [11] Dorri, A., Kanhere, S.S., Jurdak, R. (2018). Multi-agent systems: A survey. *IEEE Access*, 6: 28573-28593. <https://doi.org/10.1109/ACCESS.2018.2831228>
- [12] Selmi, A., Brahmi, Z., Gammoudi, M. (2014). Multi-agent recommender system: State of the art. In *Proceedings of the 16th International Conference on Information and Communications Security*.
- [13] Sebastia, L., Giret, A., Garcia, I. (2010). A multi agent architecture for tourism recommendation. *Trends in Practical Applications of Agents and Multiagent Systems*, 71: 547-554. https://doi.org/10.1007/978-3-642-12433-4_64
- [14] Bedi, P., Agarwal, S.K., Jindal, V., Richa. (2014). MARST: Multi-agent recommender system for e-tourism using reputation based collaborative filtering. *Databases in Networked Information Systems*, 8381: 189-201. https://doi.org/10.1007/978-3-319-05693-7_12
- [15] Cavallaro, C., Verga, G., Tramontana, E., Muscato, O. (2019). Multi-agent architecture for point of interest detection and recommendation. In *Workshop from Objects to Agents, Italy*, pp. 98-104.
- [16] Mansour, A.M., Obeidat, M.A., Hawashin, B. (2023). A novel multi agent recommender system for user interests extraction. *Cluster Computing*, 26(2): 1353-1362. <https://doi.org/10.1007/s10586-022-03655-7>
- [17] Kaur, H., Kaur, H., Singh, A. (2019). Multi-agent based recommender system for Netflix. In *Proceedings of International Conference on IoT Inclusive Life (ICIIL 2019), NITTTR Chandigarh, India*, pp. 211-221. https://doi.org/10.1007/978-981-15-3020-3_20
- [18] Alhejaili, A., Fatima, S. (2021). Multi-agent recommender system. In *2019 International Workshop on Agent-Based Complex Automated Negotiation, Macao*, pp. 103-119. https://doi.org/10.1007/978-981-16-0471-3_7
- [19] Alotaibi, B., Alotaibi, M. (2021). Consensus and majority vote feature selection methods and a detection technique for web phishing. *Journal of Ambient Intelligence and Humanized Computing*, 12(7): 717-727. <https://doi.org/10.1007/s12652-020-02054-3>
- [20] Lansdowne, Z.F., Woodward, B.S. (1996). Applying the Borda ranking method. *Air Force Journal of Logistics*, 20(2): 27-29.
- [21] McLean, I. (1990). The Borda and Condorcet principles: Three medieval applications. *Social Choice and Welfare*, 7(2): 99-108. <https://doi.org/10.1007/BF01560577>
- [22] Sutriawan, Sasoko, W.H., Alamin, Z., Ritzkal. (2025). Benchmarking text embedding models for multi-dataset semantic textual similarity: A machine learning-based evaluation framework. *Acadlore Transactions on AI and Machine Learning*, 4(2): 82-96. <https://doi.org/10.56578/ataiml040202>
- [23] Wang, H., Shen, Z., Jiang, S., Sun, G., Zhang, R.J. (2021). User-based collaborative filtering algorithm design and implementation. *Journal of Physics: Conference Series*, 1757(1): 012168. <https://doi.org/10.1088/1742-6596/1757/1/012168>
- [24] Hossain, A.K., Tasnim, Z., Hoque, S., Rahman, M.A. (2021). 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. <https://doi.org/10.61797/ijaauml.v2i1.55>
- [25] Zhang, J.D., Chow, C.Y. (2015). 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, Chile*, pp. 443-452. <https://doi.org/10.1145/2766462.276771>
- [26] Bag, S., Kumar, S.K., Tiwari, M.K. (2019). An efficient recommendation generation using relevant Jaccard similarity. *Information Sciences*, 483: 53-64. <https://doi.org/10.1016/j.ins.2019.01.023>
- [27] Liu, Y., Yang, H., Sun, G.X., Bin, S. (2020). Collaborative filtering recommendation algorithm based on multi-relationship social network. *Ingénierie des Systèmes d'Information*, 25(3): 359-364. <https://doi.org/10.18280/isi.250310>
- [28] Wang, P., Fu, Y., Liu, P., Zhu, B., Wang, F., Pamucar, D. (2024). Evaluation of ecological governance in the Yellow River basin based on Uninorm combination weight and MULTIMOORA-Borda method. *Expert Systems with Applications*, 235(1): 121227. <https://doi.org/10.1016/j.eswa.2023.121227>
- [29] Li, Z., Liu, A., Xia, L., Cao, Y., Wang, H. (2023). Differentially private Condorcet voting. In *Proceeding of the AAAI Conference on Artificial Intelligence, USA*, pp. 5755-5763. <https://doi.org/10.1609/aaai.v37i5.25714>
- [30] Hadjhenni, M., Dennouni, N., Slama, Z. (2024). 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. <https://doi.org/10.12785/ijcds/150148>
- [31] Pampin, H.J.C., Jerbi, H., O'Mahony, M.P. (2015). Evaluating the relative performance of collaborative

filtering recommender systems. *Journal of Universal Computer Science*, 21(13): 1849-1868.

[32] Lu, Y.S., Huang, J.L. (2020). GLR: A graph-based latent representation model for successive POI recommendation. *Future Generation Computer Systems*, 102: 230-244. <https://doi.org/10.1016/J.FUTURE.2019.07.074>

[33] Chi, Y.H., Zhang, P.J. (2026). DeCo-adapter: Enhancing zero-shot robustness via decoupled negative semantic suppression. *Information Dynamics and Applications*, 5(1): 1-9. <https://doi.org/10.56578/ida050101>

NOMENCLATURE

POI	Point of Interest
MCMAPRA	Model of Conciliation Multi-Agent of POI Recommendation Algorithms
RS	Recommendation System
CF	Collaborative Fettering
CBF	Content based Filtering
HF	Hybrid Filtering
MAS	Multi-Agent System
MARS	Multi-Agent Recommendation Systems
ICA	Information Center Agent
KNN	K-Nearest Neighbors
BSC	Borda Score Calculation
NV	Nevada