People's Democratic Republic of Algeria
Ministry of Higher Education and Scientific Research
Hassiba Benbouali University of Chlef
Faculty of Exact Science and Informatics
Department of informatics

THESIS

Submitted in Candidacy for the Degree of Doctorate in Third Cycle

Field : Computer systems
Speciality : Computer systems

Presented by

Djelloul BETTACHE

Title :

POI recommendation system : Formalization and evaluation

Defended on : 03/07/2025, Before the jury composed of :

ARIDJ Mohamed	MCA	Hassiba Benbouali University of Chlef	President
DENNOUNI Nassim	MCA	Higher School of Management - Tlemcen	Supervisor
HARBOUCHE Ahmed	MCA	Hassiba Benbouali University of Chlef	Co-Supervisor
MEZZOUDJ Freha	MCA	National Polytechnic School of Oran	Examiner
LOUAZANI Ahmed	MCA	University of Blida1	Examiner
TAHRAOUI Mohamed Amin	MCA	Hassiba Benbouali University of Chlef	Examiner

UHBC 2019 - 2025

Acknowledgement

First, I praise Allah, the Almighty, whose grace and wisdom have made this work possible. Second, I would like to sincerely thank my supervisor, Mr. DENNOUNI Nassim, for his guidance, patience and motivation. His invaluable advice was crucial in perfecting my contributions. I would also like to thank my co-supervisor Mr. HARBOUCHE Ahmed, the Dean of Faculty of exact sciences and informatics Mr. TAHAR ABBES Monir, the director of the LME laboratory Mr. BELALIA Azzedine for their unfailing support. I would like to extend my sincere thanks to the jury members, Mr. ARIDJ Mohamed, Mr. TAHRAOUI Mohamed Amin, Mr. LOUAZANI Ahmed, and MEZZOUDJ Freha, for the honor they have bestowed upon me by dedicating their time to read, evaluate, and provide invaluable advice on this work. Their contributions have been significant not only to me but to all present on this Scientific Advisory Board.

I would also like to thank the doctoral students for their assistance. My colleague ZOUATENIA M'hamed for their encouragement and support. MHAMMEDI TAEIB Nabil, DRAHMOUNE Ali, CHERID Ali and DADDI Abdellah for their industrial knowledge and advices about my career path.

Last but not least, I would like to express my deepest gratitude to my family for their encouragement, support and love. I feel truly grateful to my wife from the bottom of my heart, for her tremendous support and company.

Finally, I would like to thank everyone who has contributed in any way to the realization of this work.

Abstract

The rise of location-based social networks (LBSNs) such as Foursquare, Facebook, and Geolife has revolutionized user interactions by capturing rich data such as check-ins, preferences, and movement patterns. In this context, point-of-interest (POI) recommendation systems (RS) play a crucial role in guiding users to relevant locations. However, traditional approaches, particularly collaborative filtering, have limitations in addressing the complexity of spatial, social, and temporal user behaviors. This thesis primarily addresses the inability of classical similarity measures to capture contextual proximity. Three major contributions are proposed: (1) the SPPUR model, which introduces a novel similarity measure inspired by the TF-IDF method, combining trajectory sequences with the geographic proximity between users, (2) the IPUMC model, which integrates implicit check-in similarity with an explicit measure of geographic distance, and (3) the IUPJS model, based on the Jaccard index and enhanced with a spatial component derived from users' start and end points. Empirical evaluations on Foursquare datasets (New York and Tokyo) confirm the superiority of these three models over existing approaches and highlight the importance of integrating contextual factors such as location, visit order, and social relationships into POI recommendation systems.

Keywords : LBSN, POI, RS, Collaborative Filtering, Check-ins, Similarity Measures, Geographic Proximity, Trajectory Sequences, Jaccard Index

Abstract in Arabe (ملخص)

أدى ظهور الشبكات الاجتماعية القائمة على الموقع الجغرافي مثل فورسكوار، فيسوك، جيولايف إلى إحداث ثورة في تفاعلات المستخدمين من خلال التقاط بيانات غنية مثل تسجيلات الدخول والتفضيلات وأنماط الحركة. وفي هذا السياق، تلعب أنظمة توصية الأماكن ذات الأهمية دورًا حاسمًا في توجيه المستخدمين إلى المواقع ذات الأهمية. ومع ذلك، فإن الأساليب التقليدية، لا سيما الفلترة التعاونية، لها قيود في معالجة التعقيدات المكانية والاجتماعية والزمنية لسلوكيات المستخدمين. تعالج هذه الأطروحة في المقام الأول عدم قدرة مقاييس التشابه الكلاسيكية على فهم مدى التشابه في السياق. من خلال هذا المذكرة تم اقتراح ثلاث مساهمات رئيسية: (1) نموذج SPPUR، الذي يقدم مقياس تشابه جديد مستوحى من مقياس TF-IDF، يجمع بين تسلسلات المسار والقرب الجغرافي بين المستخدمين. (2) نموذج IPUMC، الذي يدمج التشابه الضمني لتسجيل الوصول مع قياس مباشر للمسافة الجغرافية. (3) نموذج IUPJS، الذي يعتمد على مؤشر جاكارد وتم تعزيزه بواسطة العنصر المكاني المستمد من نقاط بداية ونهاية زيارة المستخدمين. تؤكد التقييمات التجريبية على مجموعات بيانات فورسكوار لمدينتي (نيويورك وطوكيو) تفوق هذه النماذج الثلاثة على النماذج التي تعتمد على مقاييس التشابه الكلاسيكية، كما تسلط الضوء على أهمية دمج العوامل السياقية - مثل ترتيب المواقع خلال الزيارة والعلاقات الاجتماعية - في أنظمة التوصية الخاص بالمواقع المهمة.

الكلمات المفتاحية: تحديد الموقع الجغرافي، المواقع المهمة، أنظمة التوصية، التوصية التعاونية، تسجيلات الدخول، مقاييس التشابه، القرب الجغرافي، تسلسل المسارات، مؤشر جاكارد.

Abstract in French

Avec le développement des réseaux sociaux géolocalisés (LBSN) tels que Foursquare, Facebook et Geolife, les interactions avec les utilisateurs ont été révolutionnées par la collecte de données très riches telles que les check-ins, les préférences et les habitudes de déplacement. Dans ce contexte, les systèmes de recommandation de points d'intérêt (POI) jouent un rôle crucial en guidant les utilisateurs vers des lieux pertinents. Cependant, les approches traditionnelles, en particulier le filtrage collaboratif, ont des limites dans la prise en compte de la complexité des comportements spatiaux, sociaux et temporels des utilisateurs. Cette thèse aborde principalement l'incapacité des mesures de similarité classiques à capturer la proximité contextuelle. Trois contributions majeures sont proposées : (1) le modèle SPPUR, qui introduit une nouvelle mesure de similarité inspirée de la méthode TF-IDF, combinant des séquences de trajectoires avec la proximité géographique entre les utilisateurs, (2) le modèle IPUMC, qui intègre la similarité d'enregistrement implicite avec une mesure explicite de la distance géographique, et (3) le modèle IUPJS, basé sur l'indice de Jaccard et enrichi d'une composante spatiale dérivée des points de départ et d'arrivée des utilisateurs. Des évaluations empiriques sur des ensembles de données Foursquare (New York et Tokyo) confirment la supériorité de ces trois modèles sur les approches existantes et soulignent l'importance d'intégrer des facteurs contextuels - tels que la localisation, l'ordre des visites et les relations sociales - dans les systèmes de recommandation de points d'intérêt.

Mots-clés : LBSN, POI, RS, filtrage collaboratif, check-ins, mesures de similarité, proximité géographique, séquences de trajectoires, indice de Jaccard.

My Scientific Contributions

A. Publications

1. **Bettache, D.**, Dennouni, N. and Harbouche, A. 2025. *Improving POI Recommendation through Collaborative Filtering by incorporating the Geographical Factor into the Similarity Calculation.* Engineering, Technology & Applied Science Research. 15, 3 (**Jun. 2025**), 23629–23634. DOI: https://doi.org/10.48084/etasr.10660. **Link** : https://etasr.com/index.php/ETASR/article/view/10660

2. **BETTACHE Djelloul**, NASSIM Dennouni. *Toward a New Similarity Measure Based on Combining Tourist Check-ins and Their Trip Path for a Point-Of-Interest Recommendations in a LBSN.* International Journal of Computing and Digital Systems, **2025**, vol. 17, no 1, p. 1-13. DOI: http://dx.doi.org/10.12785/ijcds/1571111340

3. **Djelloul BETTACHE**, Nassim DENNOUNI, Ahmed HARBOUCHE. *Enhancing Points of Interest Recommendation by Integrating Users' Proximity into the Calculation of their Similarities.* Indonesian Journal of Electrical Engineering and Computer Science. **2025**. (Accepted)

B. International Conferences

1. **BETTACHE Djelloul**, NASSIM Dennouni, (**2025**). *The Impact of Neighborhood Size Used in User-User Similarity Calculation on POI Recommendation Accuracy.* First International Conference on Artificial Intelligence, Smart Technologies and Communications (AISTC'2025), 14 and 15 April, 2025 Chlef, Algeria.

2. **Djelloul BETTACHE**, Nassim DENNOUNI, Mhamed HADJ HENNI, Sarah MEDJROUD, (**2024**). *Exploring the Impact of Similarity Measures on Implicit Collaborative Filtering in Point Of Interest Recommender Systems.* In International Conference of the African Federation of Operational Research Societies (AFROS 2024). 3 to 5 November, 2024 Tlemcen, Algeria.

3. Henni, M. H., Dennouni, N., Slama, Z., Medjroud, S., & **Bettache, D**. (**2022**). *Towards an approach for online evaluation of new variants of content-based POI recommender systems by mobile tourists.* In 2022 First International Conference on Big Data, IoT, Web Intelligence and Applications (BIWA) (pp. 89-94). IEEE.

4. Medjroud, S., Dennouni, N., Henni, M. H., & **Bettache, D**. (2022). *Towards a new POI Recommendation Approach based on Implicit Trust between users.* In 2022 First International Conference on Big Data, IoT, Web Intelligence and Applications (BIWA) (pp. 19-24). IEEE.

5. **BETTACHE Djelloul**, NASSIM Dennouni, (**2022**). *Un Algorithme de filtrage collaboratif basé sur la similarité de Jaccard pour la recommandation de POI.* In Fourth edition of the international conference on research in applied mathematics and computer science (ICRAMCS 2022). March 24-26, 2022, Casablanca, Morocco.

C. National Conferences

1. **Djelloul BETTACHE**, Nassim DENNOUNI, (**2023**). *The influence of the Similarity Measures in the Recommendation System.* In The First National Conference on Educational Technologies and Informatics (CNETI'2023), Guelma, Algeria, from October 3 to 4, 2023.

2. **Djelloul BETTACHE**, Nassim DENNOUNI, (**2023**). *Recommendation Systems: Techniques, Challenges and Applications.* In National Conference on Artificial Intelligence and its Applications (NCAIA 2023). The Higher school of management, Tlemcen, Algeria. 7 december 2023.

Contents

List of Tables

List of Figures

List of Abbreviations

RS	Recommendation Systeme
CF	Collaborative Filtering
POI	Point of interest
LBSN	Location-Based Social Networks
SPPUR	Similarity of Paths and Proximity of Users for Recommending POIs
IPUMC	Integrating Proximity of Users in Modified Cosine similarity
IUPJS	Implicit User Proximity and Jaccard-based Similarity
MF	Matrix factorization
MAP	Mean Average Precision
IDCG	Ideal Discounted Cumulative Gain
NDCG	Normalized Discounted Cumulative Gain
MAE	Mean Absolute Error
RMSE	Root Mean Squared Error
TF	Term Frequency
IDF	Inverse Document Frequency
SVD	Singular Value Decomposition
ALS	Alternating Least Squares
SGD	Stochastic Gradient Descent
CBF	Content-Based Filtering
VAE	Variational Autoencoder
GNN	Graph Neural Networks

SASRec Self-Attentive Sequential Recommendation

CARS Context-Aware Recommender Systems

IR information retrieval

IoT Internet of Things

BayMAN Bayes-enhanced Multi-view Attention Networks

SVD Singular Value Decomposition

ALS Alternating Least Squares

SGD Stochastic Gradient Descent

GPS Global Positioning System

RNN Recurrent Neural Network

KNN K-Nearest Neighbors

TARE Event-Based Probabilistic Embedding

TCF Trust-enhanced Collaborative Filtering

MERec Meta-learning Enhanced POI Recommendation

CNNs Convolutional NNural networks

VAEs Variational Autoencoders

NYC New York City

General Introduction

Context

The rise of Location-Based Social Networks (LBSNs), such as Foursquare, Facebook, or Geolife, has significantly transformed how users interact with their environment and share experiences. These platforms enable the detailed capture of user behaviors, including check-ins, preferences, and movements. In this context, Point of Interest (POI) recommendation systems play a crucial role in delivering personalized services, especially in complex urban environments where the number of available POIs is continuously increasing. Among the most widely used approaches are Collaborative Filtering (CF) techniques, which rely on analyzing past user interactions to generate recommendations.

However, traditional CF methods exhibit significant limitations in the context of LBSNs. These approaches struggle to capture the complexities of interactions inherent in these networks, particularly when geographic, temporal, and social factors influence user decisions. For instance, users tend to visit places close to their current or previous location, and their choices are often influenced by their social network's behavior. These essential dimensions remain largely underutilized in existing models.

Problem Statement

POI recommendation systems in LBSNs face several major challenges. First, the data is often sparse, as users only visit a small fraction of the available POIs. Second, the cold start problem limits the ability of models to generate relevant recommendations for new users or POIs. Third, traditional similarity measures, such as Pearson correlation, cosine similarity, or Euclidean distance, fail to incorporate crucial contextual dimensions, such as geographic proximity and social influence.

Thus, the central question of this thesis is: How can we design a similarity measure that simultaneously captures behavioral and geographic similarities between users in order to provide accurate, data-scarcity-robust POI recommendations that are tailored to complex urban environments?

Motivation

This research is motivated by the rapid evolution of LBSNs and the growing importance of recommendation systems across various domains, including smart tourism, urban resource management, and enhancing user experience. The limitations of traditional approaches justify the urgent need to develop innovative models that better reflect the real dynamics of users. Integrating dimensions such as geographic proximity, visit order, and social relationships would not only improve the relevance of recommendations but also meet user expectations in real-world complex scenarios.

Contributions

This thesis makes three key contributions to address the identified challenges:

1. **The SPPUR Model**: We introduce a new similarity measure, called SPPUR (Similarity of Paths and Proximity of Users for Recommending POIs), which combines user path similarity with geographic proximity. This measure is inspired by the TF-IDF technique and takes into account visit sequences as well as the start and end points of trips. Experimental validation on real-world datasets from Foursquare (New York and Tokyo) demonstrates the superiority of SPPUR over traditional approaches in terms of PRECISION, RECALL, MAP, and NDCG.

2. **The IPUMC Model**: We propose a second model, IPUMC (Implicit Proximity and User Matching via Check-ins), which explicitly integrates geographic proximity and implicit similarity derived from user check-ins. This model relies

on a hybrid combination of cosine similarity and a geographic distance measure (GeoDist). Experimental results confirm the effectiveness of IPUMC in terms of precision, recall, F1-score, MAP, and NDCG.

3. **The IUPJS Model**: Finally, we present the IUPJS (Implicit User Proximity and Jaccard-based Similarity) model, which combines the Jaccard index for measuring the overlap of visited POIs with a geographic component based on initial and final check-ins. This approach highlights the importance of integrating spatial and behavioral dimensions to enhance recommendation quality.

These three contributions aim to incorporate essential contextual dimensions, such as geographic proximity, visit order, and social relationships, to rethink POI recommendation systems. They are validated using real-world data from Foursquare, demonstrating their superiority over traditional approaches. These works enable the development of more precise, contextualized recommendation systems that are better suited to user needs in complex urban environments.

Document Organization

This thesis follows a logical progression that lays the foundation of recommendation systems before proposing, developing, and evaluating innovative models applied to POI recommendation in location-based social networks (LBSNs).

Chapter 1 introduces the **fundamental principles of recommendation systems**, discussing various approaches (collaborative filtering, content-based, contextual, and deep learning-based), as well as strategies and evaluation metrics used to measure their performance.

Chapter 2 is dedicated to **POI recommendation in LBSNs**, defining key concepts, associated challenges (such as sparsity, cold start, and spatio-temporal complexity), and presenting a detailed literature review of existing approaches.

Chapter 3 introduces the **SPPUR model**, which leverages both user check-ins and trajectories to improve recommendation accuracy. It proposes a new similarity measure and evaluates it through various experiments.

Chapter 4 presents the **IPUMC model**, an approach that combines user similarity, calculated from their check-ins, with the notion of spatio-temporal proximity. This chapter details the construction of the similarity measure, the model architecture, the experimental protocol, and the results obtained.

Finally, **Chapter 5** introduces the **IUPJS model**, which innovatively integrates location into the calculation of user similarity. After formalizing the problem, this chapter describes the construction and prediction steps of the model and provides a comprehensive experimental evaluation based on real-world data.

PART ONE

LITERETURE REVIEW

Chaptre 1 Recommendation Systems

1.1 Introduction

This chapter provides an overview of the various recommendation approaches, beginning with a general definition of Recommender Systems (RS) and a classification based on the main methods employed: collaborative filtering, content-based filtering, context-aware recommendation, and more recent approaches based on deep learning. Each category is examined in detail, focusing on its theoretical foundations, advantages, and limitations. In addition, a section is also devoted to the evaluation of recommender systems. This evaluation process is based on several strategies (offline, online, and user studies). It use various metrics (precision, recall, NDCG, etc.) to assess the performance and relevance of the recommendations generated.

1.2 Definition of recommendation systems

Recommendation systems can be defined in several ways, given the diversity of classifications proposed for these systems, but there is a general definition by Robin Burke [1] which defines them as follows: "*Systems capable of providing personalized recommendations to guide the user towards interesting and useful resources within a large data space.*"

The two basic entities that appear in all recommendation systems are the ***user*** and the ***item***. The "*user*" is the person using a recommendation system, providing opinions on various items and receiving new recommendations from the system. The "*item*" is the general term used to refer to what the system recommends to users [2].

Recommendation systems (RS) are algorithms designed to predict a user's preferences and propose relevant items (e.g., movies, products, Points of Interest (POI), etc.) in response to information overload (see Table 1.1 below). Their origins date back to the

1990s, with early pioneers such as Tapestry (the first collaborative filter) and GroupLens, who laid the foundation for modern methods [3]. Since then, RS has adapted to various contexts, from movie recommendations (e.g., Netflix Prize) to POI suggestions (e.g., Google Maps, Foursquare). The primary goal is to maximize user utility (relevance, diversity of suggestions) and business value (e.g., time spent at a POI). An RS typically relies on three fundamental sets:

- **Users (U) :** representing individuals;
- **Objects (I) :** designating items to be recommended (POI, products, etc.) ;
- **Rating matrix (R)** : where each element $r_{u,i}$ corresponds to the explicit (rating) or implicit (click, visit duration) preference of user u for item i.

The algorithm aims to estimate a predicted rating $\widehat{r_{u,i}} = f(u, i \mid \theta)$ where θ represents the model parameters, exploiting both past interactions and contextual metadata. For example, Netflix recommends movies based on ratings and viewing history, while Spotify generates personalized playlists based on user listening and behavior.

Table 1.1 SR field of application

Domain	**Application**
E-commerce	Product recommendations
Streaming	Video/music suggestions
Health	Treatment personalization
Social Media	Recommended friends/posts

1.3 Key Features of Recommendation Systems

While academic research in recommender systems often focuses on the accuracy of **rating prediction**, aiming to minimize errors between predicted and actual user ratings, priorities in industrial environments differ considerably [4]. In real-world applications, the **practical value of the recommendations themselves** takes precedence over precise score estimation. The usefulness, relevance, and engagement potential of the recommended items are considered far more important from a business and user experience perspective. In industrial contexts, recommendation systems are typically expected to fulfill four essential functions [5].

a. Help Users Find Relevant Items : help users discover items that match their current preferences or needs, especially when faced with too many choices or unfamiliar content.

b. Facilitate Item Exploration : encourage users to explore the entire product list, including items they might not consider otherwise, thus increasing diversity and serendipity.

c. Helping users make decisions : help users make better decisions by providing relevant suggestions, contextual information or user-generated feedback (e.g. ratings, reviews).

d. Improve user satisfaction : increase time spent on the platform and foster long-term loyalty by providing relevant, personalized recommendations.

1.4 Classification of Recommendation Systems

Several classifications of recommendation systems have been identified in the literature, each offering a distinct perspective on how recommendations are generated and organized. Below are some of the most widely recognized classifications.

1.4.1 The classical classification

This classification of Adomavicius and Tuzhilin [6] categorizes recommendation systems into three primary approaches:

- **Collaborative filtering**, which relies on the preferences or behaviors of a group of users to recommend items to others with similar interests.

• **Content-based filtering**, which uses item features and user profiles to suggest similar items to those previously liked or interacted with.

• **Hybrid filtering**, which combines collaborative and content-based methods to leverage the strengths of both approaches and mitigate their limitations (such as the cold-start problem).

1.4.2 The classification proposed by Su and Khoshgoftaar

This classification focuses specifically on collaborative filtering. This classification further breaks down collaborative methods into two categories [7].

• **Memory-based techniques**, which use user-item interactions stored in a matrix to find similarities.

• **Model-based techniques**, which use machine learning models to predict user preferences based on historical data.

This typology emphasizes the mechanics and data dependency of collaborative filtering methods, highlighting how similarity measures and learning algorithms influence performance.

1.4.3 The classification introduced by Rao

This classification Classifies recommendation systems based on the type and source of data used [8]. These may include:

- **Explicit data**, such as user ratings or reviews.
- **Implicit data**, such as user clicks, views, or purchase history.
- **Contextual data**, including time, location, device, or demographic information.

This classification underlines the importance of data diversity and contextualization in tailoring recommendations to individual users and real-world environments.

The Figure 1.1 below summarizes the different types of classification of recommender systems.

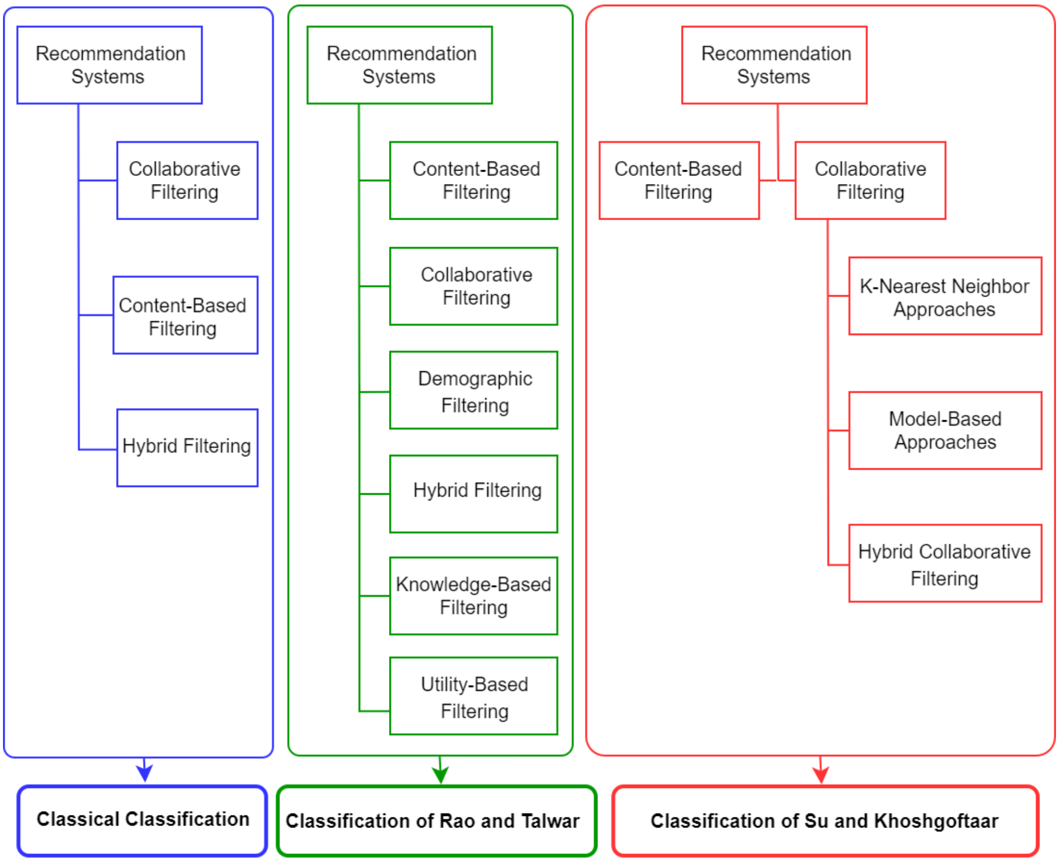

Figure 1.1 Principale classification of RS

1.5 Recommendation Systems techniques

Recommendation systems (RS) are characterized by their methods of modeling preferences and interactions. Their classification mirrors the evolution of techniques, ranging from simple statistical approaches to complex deep learning architectures.

1.5.1 Collaborative Filtering (CF)

Collaborative filtering is a widely used technique in recommendation systems that relies on the preferences or behaviors of users to suggest items. It is primarily divided into two main approaches: model-based and memory-based methods [9] [10]. Model-based collaborative filtering employs machine learning algorithms to predict user preferences. Common techniques include clustering algorithms to group similar users or items, association rule mining to uncover frequent itemsets, Bayesian networks to model probabilistic relationships, and neural networks for capturing complex, non-linear interactions. On the other hand, memory-based collaborative filtering directly uses historical user-item interactions without building a predictive model [11]. It includes user-based methods, which find similar users to generate recommendations [12], and item-based methods, which identify similar items based on user ratings [13]. The Figure 1.2 below illustrates the different techniques used in collaborative filtering

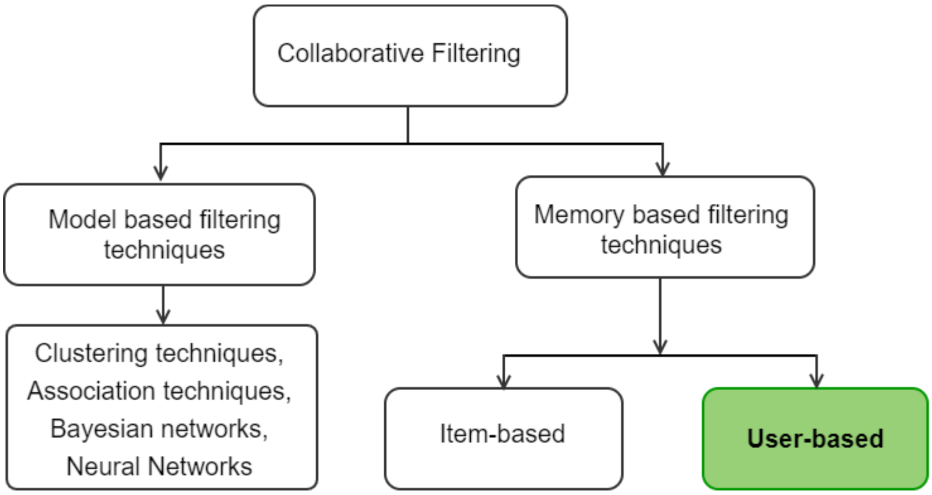

Figure 1.2 Collaborative filtering (CF) techniques

Collaborative filtering (CF) is a fundamental method in recommendation systems, based on the similarity algorithms:

- Similar users tend to prefer the same items.
- Similar items are often appreciated by the same users.

1.5.1.1 General formalization

Let:

- $U = \{u_1, u_2, \ldots, u_m\}$ be the set of users,
- $I = \{i_1, i_2, \ldots, i_n\}$ be the set of items (movies, products, points of interest, etc.)
- $R \in \mathbb{R}^{m \times n}$ be the rating matrix, where $r_{ui} \in \mathbb{R}$ represents the rating given by user u to item i (or an implicit interaction measure).

The goal of collaborative filtering is to estimate a prediction $\widehat{r_{ui}}$ for unseen user-item pairs (u, i), i.e., those for which r_{ui} is unknown.

1.5.1.2 Memory-based collaborative filtering

In memory-based collaboration, user ratings for items stored in the system are used directly to predict ratings for unrated items. This type of filtering passes through two important phases: the calculation of similarity and the calculation of rating prediction. Similarity is calculated either for all users or for all items [14]. Figure 1.3 summarizes memory-based collaborative filtering.

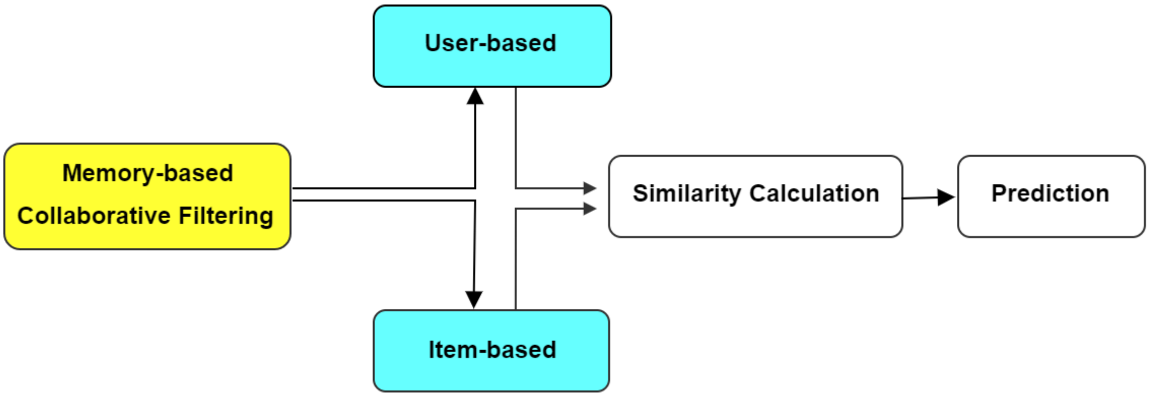

Figure 1.3 Memory-based collaborative filtering

A. User-Based collaborative filtering

In this approach, the goal is to predict the preferences of user u by identifying a set of users similar to u, referred to as neighbors, and then combining their ratings of item i. This enables the recommendation of items to user u that have been highly rated by users similar to u [15]. To achieve this objective, two steps are necessary:

<u>Step one</u> : Calculation of similarity between users:

The similarity between two users u and v is measured using statistical metrics. The most common ones are cosine similarity and Pearson correlation [16].

1) Cosine similarity:

Cosine similarity measures the cosine of the angle between two non-zero vectors in a multidimensional space. It is widely used in text mining and recommendation systems to assess the orientation of user or item vectors [17].

$$\text{sim}(u,v) = \frac{\sum_{i \in I_{uv}} r_{u,i} \cdot r_{v,i}}{\sqrt{\sum_{i \in I_{uv}} r_{u,i}^2} \cdot \sqrt{\sum_{i \in I_{uv}} r_{v,i}^2}} \tag{1.1}$$

2) Pearson correlation:

Pearson correlation evaluates the linear relationship between two variables.

$$\text{sim}(u,v) = \frac{\sum_{i \in I_{uv}} (r_{u,i} - \overline{r_u})(r_{v,i} - \overline{r_v})}{\sqrt{\sum_{i \in I_{uv}} (r_{u,i} - \overline{r_u})^2} \cdot \sqrt{\sum_{i \in I_{uv}} (r_{v,i} - \overline{r_v})^2}} \tag{1.2}$$

Where,

- $r_{u,i}$ represents the interaction (rating, click, time spent) of user u with item i.
- I_{uv} is the set of items evaluated by both users u and v.

<u>Step two</u> : Prediction of the rating

The predicted rating for user u and item i is expressed as :

$$\widehat{r_{u,i}} = \overline{r_u} + \frac{\sum_{v \in N_u(i)} \text{sim}(u,v) \cdot (r_{v,i} - \overline{r_v})}{\sum_{v \in N_u(i)} |\text{sim}(u,v)|} \tag{1.3}$$

Where,

- r_u represents the average rating of user u.
- $N_u(i)$ is the set of the K users most similar to u who have rated item i.

Discussions

Collaborative filtering based on user similarity operates by identifying profiles with comparable behaviors. For instance, if Fatima and Ali have visited the same Points of Interest (POI) 80% of the time, their cosine similarity will be high. In this case, a POI visited by Ali but not by Fatima can be recommended to her.

However, these approaches, whether based on user or item similarities, face several limitations. First, scalability presents a major challenge: computing similarities for millions of users or items requires substantial computational resources, both in terms of time and memory. Second, data sparsity diminishes the reliability of recommendations, as most user-item interactions (ru,i) are missing, which leads to less accurate similarity estimations. Finally, these methods are vulnerable to the cold start problem: new users or items, lacking historical interaction data, cannot be effectively integrated into the system. These constraints limit the performance of traditional collaborative filtering methods, especially in large-scale or sparse data contexts.

B. Item-Based collaborative filtering

The Item-Based approach is based on the following idea: "If a user liked item A, they are likely to enjoy items similar to A." In this case, instead of finding similar users, the focus is on identifying items that are similar to those already liked by the user [18]. The objective is to predict preferences by leveraging the similarity between items (e.g., "Users who visited this POI also liked...").

1) Calculating similarity between items

Similarly, the similarity between two items, i and j, is computed using the cosine measure,

$$\text{sim}(i,j) = \frac{\sum_{u \in U_{ij}} r_{u,i} \cdot r_{u,j}}{\sqrt{\sum_{u \in U_{ij}} r_{u,i}^2} \cdot \sqrt{\sum_{u \in U_{ij}} r_{u,j}^2}} \tag{1.4}$$

Where,

- *Uij* denotes the set of users who have rated both *i* and *j*.

2) Prediction

Prediction is expressed as :

$$\widehat{r_{u,i}} = \frac{\sum_{j \in N_i(u)} \text{sim}(i,j) \cdot r_{u,j}}{\sum_{j \in N_i(u)} |\text{sim}(i,j)|} \tag{1.5}$$

Where,

- *Ni(u)* is the set of *K* items similar to *i* that user *u* has already rated.

Discussions

Item-based collaborative filtering is founded on the principle that if two Points of Interest (POIs), such as the Louvre Museum and the Centre Pompidou, are frequently visited by the same users, they are considered to be similar. Thus, a user who has visited the Louvre would be recommended the Centre Pompidou. This approach offers several advantages: first, stability, as item similarities generally vary less than user similarities; second, efficiency, since the computational cost is lower, with items typically being fewer in number than users. However, this method also presents limitations, particularly in terms of scalability when dealing with large item catalogs and its dependency on data sparsity. When interactions are too infrequent, the computed similarities may lack reliability. Additionally, it struggles with the cold start problem, where no interactions are available for new items.

1.5.1.3 Model-based collaborative filtering

This model is the opposite of memory-based systems. Model-based systems use ratings to learn a prediction model. They then use the available data to train the model, which is then used to predict user ratings for new items [19]. The advantage is that these models can be built offline and can be quickly used to calculate online recommendations. The algorithms of this approach can be broken down into three subtypes: clustering, matrix factorization (MF) and deep learning [20]. Each of these approaches contains algorithms of different types, as shown in Figure 1.4 below.

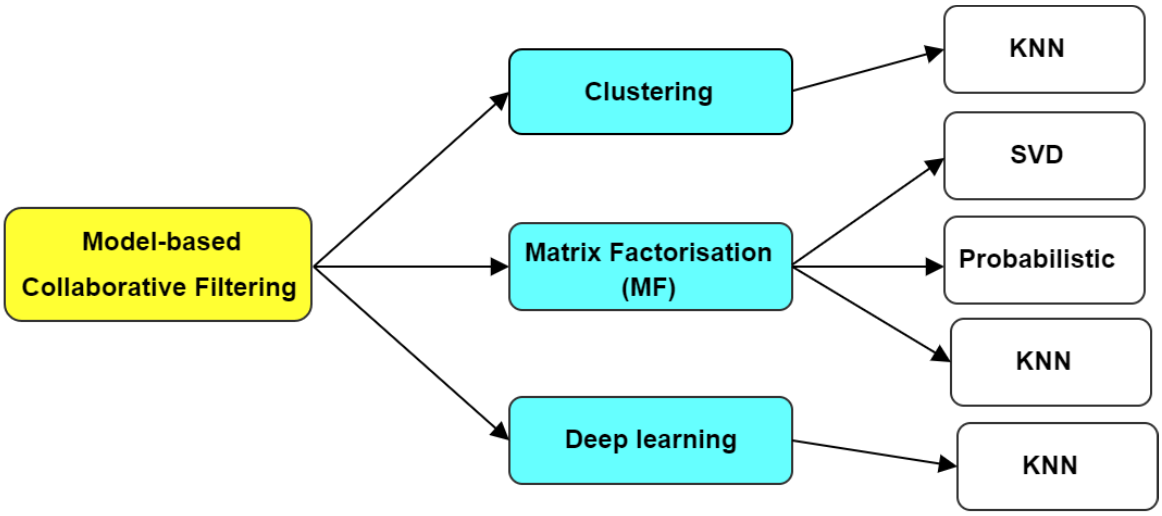

Figure 1.4 Model-based collaborative filtering

A. Matrix factorization (MF)

A powerful method for Collaborative Filtering (CF) is matrix factorization, notably introduced by Simon Funk during the Netflix Prize in 2006 [21]. This approach aims to reduce the dimensionality of the user-item matrix by identifying latent factors (hidden features) that account for the observed interactions [20, 22].

a. Formulation

The idea is to factorize the scoring matrix R $\in R^{m\times n}$ into two latent matrices:

- $U \in R^{m\times k}$ Matrix of latent user factors,
- $V \in R^{n\times k}$ Item latent factor matrix,

Such as :

$$R \approx U \cdot V^T \tag{1.6}$$

Each row of U represents a user in the k-dimensional latent space, and each row of V represents an item in the same space. For example, let $k=2$, a user u with $U_u = [0.5,1.2]$ and an item i with $V_i = [1.0,0.8]$.

The predicted score is $\widehat{r_{u,i}} = 0.5\times1.0+1.2\times0.8=1.46$.

The objective is to minimize the prediction error, often by the cost function :

$$\min_{U,V} \sum_{(u,i)\in\mathcal{K}} \left(r_{u,i} - \langle U_u, V_i\rangle\right)^2 + \lambda(|U|^2 + |V|^2) \tag{1.7}$$

Where,

- K is the set of observed user-item pairs,
- λ is a regularization parameter.

Discussions

Among the most common extensions of matrix factorization (MF) are techniques such as Singular Value Decomposition (SVD), Alternating Least Squares (ALS), and Stochastic Gradient Descent (SGD). These methods efficiently handle data sparsity by learning latent factors that capture implicit preferences, which are not directly observable from the data. Additionally, these latent dimensions can sometimes be interpreted as underlying themes, such as a preference for historical museums or outdoor locations, enhancing the interpretability of the model.

However, despite their performance, collaborative filtering (CF) approaches face several significant limitations. One such challenge is the cold-start problem, which affects new users or items that have few or no past interactions. A common solution is to combine CF with content-based approaches. Furthermore, recommendation systems are often subject to a popularity bias, where the most popular items such as highly visited tourist points of interest (POIs) are favored over lesser-known but potentially more relevant alternatives. Additionally, there is the issue of temporal dynamics: user preferences evolve over time, which static models struggle to capture without continuous adaptation.

Despite these challenges, CF remains widely used in practical applications. For instance, Netflix relies on matrix factorization to predict movie ratings; Amazon uses product similarity to generate personalized recommendations; and Foursquare leverages CF to suggest points of interest (POIs) based on check-ins from users with similar profiles.

Table 1.2 Advantages and disadvantages of FC

Advantages	**Disadvantages**
Does not require knowledge of the items	Sensitive to the cold start problem
Captures subtle and complex preferences	Matrix sparsity problem
Easy to implement	Can lead to overfitting if the model is too complex

1.5.2 Content-based filtering approaches

Content-based recommendation methods operate on a fundamental principle: user preferences can be inferred from the characteristics of the items they have previously liked [23] [24]. Unlike collaborative filtering, which analyzes collective user behaviors [25], this approach specifically examines the relationship between a user and the attributes of the content they consume.

More formally, Content-Based Filtering (CBF) represents a class of recommendation algorithms that relies on (1) the intrinsic features of items (such as metadata, categories, text, etc.) and (2) the user's individual history (their past interactions) [26].

In contrast to collaborative methods, CBF does not require data from other users. It focuses exclusively on :

1. Analyzing the properties of items (e.g., a Point of Interest (POI) "museum" tagged with "art," "culture").

2. Building a user profile from previously liked items (e.g., if a user frequently visits museums, the system will recommend locations with similar attributes).

This approach is particularly useful for (1) mitigating the "cold start" problem for new users and (2) recommending niche items that may be less popular but are relevant due to their specific characteristics.

Finally, CBF personalizes recommendations by establishing direct matches between individual preferences and the attributes of content, without relying on the behaviors of a community.

1.5.2.1 Construction of the profile

The Content-Based Filtering (CBF) approach is based on two profiles :

a. Item Profile :

This profile represents the characteristics of an item in a structured form, which includes: (1) **Textual Data** : Description of the Point of Interest (POI), e.g., "history museum," tags such as culture or architecture, (2) **Categorical Data** : Type of restaurant, such as "Italian" or "vegetarian.", and (3) **Multimedia Data** : Images (e.g., urban landscape), geographical metadata (e.g., GPS coordinates) [27].

Each item (e.g., movie, book, product, point of interest) is represented by a vector of features extracted from its explicit attributes [24]. For example:

- **Movies**: genre, actors, director, duration, synopsis.
- **Articles** : keywords, summary, category.
- **POIs (Points of Interest)** : type (restaurant, museum, etc.), location, ratings, keywords from reviews.

These attributes can be encoded as TF-IDF vectors (for text), one-hot vectors (for categories), or numerical representations learned via methods such as word embeddings (e.g., Word2Vec, BERT).

b. User Profile :

A user profile is generated by aggregating the characteristics of the items they have interacted with [28] [29]. Several approaches are used for this modeling :

1) Standard aggregation methods

- Weighted average of TF-IDF representations:

Items that the user likes are represented as TF-IDF vectors, which are then combined through a weighted average based on interaction scores.

- Dense representations (Embeddings) :

Using modern language models (e.g., Word2Vec [30], BERT [31]), the embeddings of consumed items are averaged to obtain a dense representation of the user's profile.

2) Rocchio algorithm (Incremental Update) [32]

- Profile update formula :

$Profile_u = \alpha \cdot Profile_u + \beta \cdot Item_i$

Where,

- α controls the weight of historical preferences,
- β adjusts the impact of new interactions.

3) Advanced supervised approaches

- **Classification/Regression**:

  A predictive model (e.g., SVM, logistic regression) can be trained to estimate the relevance of an item based on the user's profile characteristics [33].

1.5.2.2 The recommendation process

This process is based on the following steps:

Step 1: Item-Profile similarity calculation

To determine the most relevant items for a user, the system compares their profile with each candidate item using similarity measures.

a. Cosine similarity

This measures the angle between the user profile and item vectors:

$$\text{sim}(Profil_u, Item_i) = \frac{Profil_u \cdot Item_i}{|Profil_u| \times |Item_i|} \tag{1.8}$$

This type of similarity is ideal for textual data (e.g., TF-IDF, embeddings).

b. Euclidean distance

This calculates the "distance" between features:

$$\text{distance} = \sqrt{\sum_k \left(Profil_u(k) - Item_i(k)\right)^2} \tag{1.9}$$

The smaller the distance, the more similar the item is.

c. Neural networks

Advanced models (e.g., MLP, transformers) capture non-linear relationships between the profile and items.

Step 2: Ranking and selection

Items are ranked in descending order by similarity score. Only the top-k items (e.g., top 10) are recommended to the user. The content-based recommendation process finds practical applications across various domains, demonstrating its effectiveness in personalizing suggestions. In the cinematic domain, for instance, a user whose profile indicates a strong preference for "science-fiction" and "action" genres will be recommended films like Dune or Matrix, which share these thematic characteristics. In article recommendations, a reader who regularly engages with political content will automatically receive suggestions related to topics such as elections or foreign policy, aligning with their historical interests. Lastly, in the domain of geolocated points of interest (POIs), a user who frequently visits parks and cafés will be directed to similar locations, such as botanical gardens or cafés with terraces, showcasing the system's ability to extrapolate preferences from past interactions. These examples illustrate how analyzing the intrinsic features of items allows the generation of relevant recommendations that align with each user's profile.

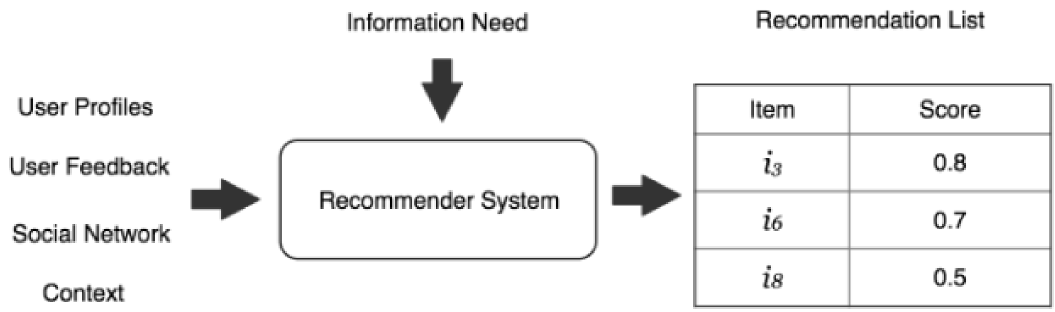

Figure 1.5 General recommendation process [34]

1.5.2.3 Advantages and disadvantages of CBF

a) Key Advantages of content-based methods :

These approaches provide a high degree of personalization by closely aligning with individual preferences, and they partially address the cold start problem by functioning effectively even with limited users or items. Their transparency allows recommendations to be explained through the characteristics of the items (e.g., tags, genres). Unlike collaborative filtering, content-based methods avoid issues related to data sparsity (missing data) and do not rely on user interactions. They are particularly well-suited for niche domains (e.g., classical music) and can recommend new items as long as their metadata is rich.

b) Main limitations :

However, these methods suffer from over-specialization, as the recommendations are often too similar to past consumption, limiting the discovery of new content. Their performance heavily depends on the quality of metadata: incomplete or noisy attributes degrade the effectiveness of the system. While content-based methods mitigate the cold start problem for new items, they remain vulnerable when applied to new users without interaction history. Additionally, the diversity of recommendations may be insufficient without complementary mechanisms (e.g., hybridization with other approaches).

In conclusion, content-based filtering excels in contexts where item attributes are rich and reliable but often requires integration with other techniques to balance relevance and diversity, as shown in the table below:

Table 1.3 strengths and weaknesses of CBF

Strength	**Weakness**
Fine personnalisation	Over specialization (lack of diversity)
Explainability of recommendations	Dependence on metadata
Robustness to cold start (items)	Sensitivity to cold start (users)
Independence from other users	Need for structured item features

1.5.2.4 Discussions

Modern recommendation systems are evolving toward a synergistic integration of content-based filtering (CBF) and other advanced paradigms. Hybrid architectures, such as LightFM, exemplify this trend by effectively combining CBF with collaborative filtering, merging both user/item embeddings and contextual features like location or time of day. Simultaneously, deep learning enhances these systems: convolutional neural networks (CNNs) enable the exploitation of unstructured data, such as point-of-interest (POI) images, while transformers (e.g., BERT) provide a nuanced understanding of user reviews and complex textual descriptions.

Despite the apparent dominance of collaborative methods, CBF retains distinctive advantages in three key areas: it excels in handling sparse data (low interaction density), remains the preferred choice for critical applications requiring high explainability (e.g., healthcare, finance), and remains indispensable for specialized content (e.g., academic research). The future of CBF appears particularly promising, especially with its integration with cutting-edge techniques like multimodal embeddings (combining text, images, and context) and semantic analysis through large language models (LLMs), thus enabling systems that are not only more contextually intelligent but also more adaptive, without compromising the transparency that defines it. This ability to evolve while maintaining its core strengths positions CBF as a lasting and indispensable component in the increasingly complex recommendation ecosystem.

1.5.3 Contextual recommendation

Contextual recommendation represents a significant extension of traditional recommendation systems by incorporating dynamic external factors, known as contexts, into the suggestion generation process. Unlike classical approaches that focus solely on user-item interactions (e.g., ratings or check-ins), this approach acknowledges that user preferences may vary substantially depending on the situation in which they find themselves. Some of the most commonly used contextual factors include:

- **User's geographical location** : This factor is crucial in Point of Interest (POI) recommendation systems, as the relevance of a recommended location often depends on its proximity. A contextual system might prioritize POIs within a limited radius around the user's current position.
- **Time of day or date** : A restaurant might be recommended for lunch if the local time is noon, or a cinema in the evening. This temporal factor allows recommendations to align with the user’s daily rhythm.
- **Weather conditions** : Intelligent systems can adjust suggestions based on the weather. For instance, in the event of rain, recommending an indoor café or museum would be more suitable than suggesting a park or an outdoor activity.

- **Crowd levels or local events** : Some systems incorporate real-time data on crowd density or nearby cultural events to offer recommendations that are in tune with the user's immediate environment.

1.5.3.1 User context modeling

Context refers to the set of external conditions that influence a user's preferences at a given moment. This can include factors such as their GPS location (spatial context), the time of day or the day of the week (temporal context), or environmental elements such as weather, lighting, or noise levels. Social context, on the other hand, considers the presence of friends or colleagues. The purpose of incorporating context is to tailor recommendations to enhance their immediate relevance, such as suggesting an umbrella when it rains or a café in the morning. Consequently, user context must encompass the following dimensions :

a. Location

This dimension enables the prioritization of geographically accessible Points of Interest (POIs), such as restaurants in the user's vicinity. For example, a user located in Chlef (Algeria) will receive café recommendations within their city rather than in Tlemcen, thus optimizing the relevance of suggestions based on their location. This dimension can be modeled, for example, by using the Euclidean distance, calculated via the formula below:

$$d(u,i) = \sqrt{(x_u - x_i)^2 + (y_u - y_i)^2} \tag{1.10}$$

Where,

- $(x_u,\ y_u)$ = user's coordinates, $(x_i,\ y_i)$ = POI coordinates. This distance can be integrated as a feature in the prediction function below:

$$\widehat{r_{u,i}} = f(\text{similarité}, d(u,i), \text{contexte}) \tag{1.11}$$

b. Time and weather

The primary goal is to personalize recommendations based on various contextual factors, such as the time of day (for instance, suggesting breakfast menus in the morning) or weather conditions (directing users towards indoor activities like museums on rainy days). This dynamic approach ensures that the suggestions remain relevant to the user's immediate context, thereby enhancing the timeliness and utility of the recommendations in real time.

To represent time in a cyclical manner and capture its periodic nature (such as morning vs. night preferences), trigonometric transformations are employed:

$$h_{sin} = si\,n\left(\frac{2\pi \cdot \text{heure}}{24}\right), \quad h_{cos} = \cos\left(\frac{2\pi \cdot \text{heure}}{24}\right) \tag{1.12}$$

The climatic conditions (rain, sunshine, etc.) are encoded as embeddings and then merged with the user's or recommended items' features. In the case of rain detected via a weather API, the system can:

- Identify cafés with covered terraces.
- Prioritize these establishments in the recommendations to provide an experience tailored to current weather conditions.

c. Social context

The recommendation system takes the user's social context into account, adapting its suggestions based on the user's company (family, colleagues, friends, etc.). This social dimension is modeled through relational graphs that exploit connections between users, such as friend networks on social platforms. For example, when a user is identified as being with family, the algorithm will prioritize locations such as parks or family-friendly restaurants, rather than Museums or places more suitable for outings with colleagues. This approach refines suggestions based on the present social circle, thereby enhancing the relevance and acceptability of the recommendations.

1.5.3.2 Context integration techniques

To improve the relevance of recommendations, the following techniques offer increased flexibility for integrating complex contextual data.

a. Tensors and multimodal factorization

This approach relies on a tensorial representation (a multidimensional matrix) of the interactions between users, items, and contexts. Formally, we define a tensor R∈Rm×n×tR∈Rm×n×t, where mm represents the users, nn the items, and tt the contextual dimensions (e.g., time of day). The factorization of this tensor (using methods like Tucker or CP) allows for the decomposition of RR into latent factors.

$$R \approx \sum_{k=1}^{K} \lambda_k \cdot U_k \circ V_k \circ C_k \tag{1.13}$$

Where Uk, Vk and Ck correspond respectively to the latent embeddings of users, items and contexts, while λk is their weighted contribution.

b. Contextual neural networks

Two main architectures are employed:

- Feature Concatenation: The embeddings of users, items, and contexts are merged through a dense layer to generate predictions.
- Transformers: Models such as BERT capture temporal dependencies by encoding action sequences (e.g., check-in histories) while incorporating their timestamps. This method enables the detailed modeling of preference evolution over time or other contextual variables.

1.5.3.3 Discussions

Contextual recommendation integrates situational dimensions to refine the suggestions made to users. Technically, this integration can be achieved either through filtering approaches (pre- or post-processing of data) or via specific models such as tensor factorizations or deep learning architectures that directly incorporate context into their prediction functions. This method dynamically adjusts suggestions based on spatiotemporal and environmental parameters, thus offering a more relevant experience, particularly for mobile and tourism applications. However, this approach faces several major challenges: algorithmic complexity due to the dimensional explosion of data, sparsity in certain specific contexts, and the need for real-time processing of fluctuating data such as geolocation or weather. Practical applications demonstrate its effectiveness, such as Google Maps, which adjusts its restaurant recommendations based on the time of day; Spotify, which personalizes playlists depending on the time; or Uber Eats, which prioritizes specific dishes based on weather conditions. Future research directions are focused on reinforcement learning for dynamic adaptation, multi-source contextual data fusion, and the development of solutions that protect user privacy. These promising technical advancements, however, must address the challenge of balancing fine-grained personalization with the protection of data privacy.

1.5.4 Deep learning approaches

Neural networks offer a more refined model for capturing the complex, non-linear interactions between users and items. For this reason, they surpass traditional approaches such as collaborative filtering or matrix factorization. Their strength lies in their ability to identify implicit relationships, handle data sparsity more effectively, and generate richer and more expressive representations [35].

1.5.4.1 Variational autoencoders

A Variational Autoencoder (VAE) is a type of neural network that learns to compress (encode) input data into a latent space and subsequently reconstruct it (decode) [36]. VAEs combine the learning of latent representations with probabilistic generation to reconstruct the user-item interaction matrix, R. This interaction matrix R (users $\times$ Points of Interest, POIs) is typically very sparse, as each user only visits a limited number of locations out of the total available [37]. To address this issue, the VAE compresses a user's preferences into a low-dimensional latent space and then attempts to reconstruct their complete interactions from this condensed representation, enabling:

- Prediction of relevant POIs by identifying unvisited places that are likely to interest the user.
- Handling uncertainty by probabilistically modeling latent variables for enhanced robustness.
- Generalization through the variational approach.

This method (VAE) is based on:

1) Encoder :

$q_{\phi}(z|u) = \mathcal{N}(\mu_u, \sigma_u^2)$ where z is the user's latent vector u.

2) Decoder :

Reconstruction of $\hat{R} = p_{\theta}(R|z)$ using a multinoulli or Gaussian distribution.

3) Cost function :

$$\mathcal{L} = \underbrace{\left|R - \hat{R}\right|_F^2}_{\text{Reconstruction}} + \underbrace{\mathrm{KL}\left(q_{\phi}(z|u)|p(z)\right)}_{\text{Regularisation}} \tag{1.14}$$

1.5.4.2 Graph neural networks (GNNs)

Graph Neural Networks (GNNs) are deep learning models that operate directly on graph structures (nodes + edges), propagating information between neighboring elements. These GNNs leverage heterogeneous graphs to transmit information between nodes (e.g., users, POIs) through edges (e.g., visits, similarities) [38].

Furthermore, GNN models find particularly relevant applications in two key areas:

A. Spatial relationships between POIs:

Architectures such as Spatial-GCN exploit geographic proximity or co-visit patterns between locations. For instance, if two POIs are frequently visited together, the model automatically learns to associate their representations, thereby improving the relevance of location-based recommendations [39].

B. Social relationships between users:

By integrating social network data (friendship links, trust relationships, or behavioral similarities), the model can propagate user preferences throughout the social graph. This approach enables the inference of implicit tastes based on user connections, thus compensating for the lack of explicit data [40].

This method (GNN) is based on:

1) Message Propagation:

$$h_i^{(l+1)} = \sigma\left(\sum_{j\in\mathcal{N}(i)} W^{(l)} \cdot h_j^{(l)}\right) \tag{1.15}$$

where *N(i)* is the neighborhood of node *i*, and *W(l)* is a weight matrix.

2) Pooling, which involves hierarchical aggregation to capture patterns at multiple scales.

1.5.4.3 Transformers

Transformers are deep learning models based on the attention mechanism, particularly effective for sequence processing tasks, such as text or trajectory analysis. This method relies on transformers that use multi-head attention to dynamically weigh the past interactions of a user [41].

In the context of recommending points of interest, sequential models utilize the user's movement history in the form of temporal sequences of check-ins or routes. The multi-head attention mechanism, which is a characteristic feature of these architectures, allows for the dynamic analysis and weighting of the relative importance of each previously visited point of interest (POI) to predict the next destination [42]. This approach has the key advantage of efficiently modeling long-term dependencies in mobility behaviors, thus capturing both recurring habits and significant exceptions.

Specialized architectures, such as SASRec (Self-Attentive Sequential Recommendation)

[43] and BERT4Rec (adaptation of transformers for recommendations) [44], have excelled in this field. They provide a high degree of flexibility in processing visit sequences while maintaining interpretability due to the explicit attention weights. These models are particularly effective in predicting future movements based on past patterns, even when these patterns span long periods or exhibit complex structures.

Transformers use multi-head attention to dynamically weigh past user interactions using mechanisms, as indicated in the formula below [45]:

$$\text{Attention}(Q, K, V) = \text{softmax}\left(\frac{QK^T}{\sqrt{d_k}}\right)V \tag{1.16}$$

where $(Q,\ K,\ V)$ are the queries, keys and projected values.

For example, this formula predicts the next visited POI by analyzing the historical movement data (e.g., "After a museum visit, the user often goes to a café") using the BERT4Rec model, which encodes check-in sequences with bidirectional Transformers.

1.5.4.4 Discussions

Variational models (VAE) stand out for their probabilistic approach, which is particularly well-suited for handling incomplete data. Their ability to manage the cold-start problem for new users makes them a robust solution for systems that frequently deal with new users. However, their reliance on the Gaussian assumption can limit their expressiveness in complex scenarios, as noted by several recent studies (e.g., [46] [47]).

Graph Neural Networks (GNN) introduce an essential relational dimension for spatial recommendation. Their strength lies in the explicit modeling of connections between entities (users-POIs, POIs-POIs), which allows for the integration of geographical or social knowledge. However, as pointed out in [48], their sensitivity to data quality and computational complexity present significant practical challenges. Transformers revolutionize sequential modeling with their attention mechanism, surpassing traditional RNNs in capturing long-term dependencies. Recent applications in POI recommendation (SASRec, BERT4Rec) demonstrate remarkable effectiveness for check-in sequences. Nevertheless, their resource-intensive nature remains a major drawback, particularly for mobile applications where energy efficiency is crucial.

The following table (Table 1.4) compares these three approaches.

Table 1.4 Summary of comparisons between VAE, GNN and tranformer models

Model	Main Objective	Advantage
VAE	Reconstruction of the user preference matrix	Compression + Bayesian regularization
GNN	Exploitation of spatial/social relationships	Explicit graph modeling
Transformers	Modeling of sequential interactions	Capturing long-range temporal dependencies

The analysis reveals promising synergies between the three architectural paradigms:

- The **VAE-GNN** combination has the potential to overcome their individual limitations. Variational Autoencoders (VAEs) can generate noise-resilient latent representations, which can then be enriched through the relational modeling capabilities of Graph Neural Networks (GNNs) [37].
- The **Transformer-GNN** integration opens innovative possibilities for jointly modeling both the temporal dimension (e.g., sequences of user visits) and the spatial dimension (e.g., networks of Points of Interest POIs) [49] [50].

Three critical research directions emerge:

a) Scalability: Hybrid architectures must address the combinatorial explosion in computational costs. Techniques such as graph sampling and sparse attention mechanisms warrant further exploration.

b) Interpretability: Bridging the inherent explainability of GNNs with the black-box nature of Transformers requires novel interpretability strategies, such as attention-based explanation models.

c) Spatial Fairness: As emphasized in [51], these complex models risk reinforcing geographic biases, necessitating the design of specific fairness-aware mechanisms and regulatory constraints.

1.5.5 Taxonomy and synthesis of recommendation systems

Recommender Systems (RSs) can be categorized into multiple paradigms, each with distinct strengths and limitations. Among the most widely used is **collaborative filtering (CF)**, which can be user-based or item-based [52].

- **User-based CF** recommends Points of Interest (POIs) by leveraging similarity metrics between users commonly cosine similarity. This approach is easy to implement and effective for active users but suffers from well-known issues such as cold-start problems, data sparsity, and limited scalability due to its algorithmic complexity.
- In contrast, **item-based CF** recommends POIs similar to those already visited (e.g., suggesting the Pompidou Centre after a visit to the Louvre). While this approach is more stable and computationally efficient, it remains sensitive to item cold starts and heavily dependent on interaction density.

Advanced extensions of CF, such as **matrix factorization techniques** (SVD, ALS, SGD), enable the discovery of latent factors representing implicit user preferences (e.g., affinity for museums or parks) [22]. However, these methods introduce new challenges, including popularity bias and difficulties in modeling the temporal evolution of preferences. Such approaches are commonly employed in real-world systems like Netflix, Amazon, and Foursquare.

To mitigate the limitations of traditional CF, **hybrid methods** have been proposed, integrating CF with **content-based filtering (CBF)** [53]. These models simultaneously utilize user–POI interactions and contextual attributes such as location, time, or weather conditions. Algorithms like **LightFM** exemplify this synergy by combining latent embeddings with explicit feature representations [54]. The advent of

deep learning has further enhanced hybrid systems: **convolutional neural networks (CNNs)** are used to analyze POI-related images, while **transformers** such as BERT enable advanced processing of textual reviews.

Furthermore, **context-aware recommender systems (CARS)** dynamically adapt recommendations by incorporating situational variables (e.g., time, location, weather) [55]. This is achieved using techniques like pre-/post-filtering, **tensor factorization**, and adaptive deep models. Such methods are prominently used in applications like Google Maps, Spotify, and Uber Eats, where context significantly affects recommendation relevance. Nonetheless, these systems must address key challenges, including dimensionality explosion, real-time processing, and privacy protection.

Advanced RS architectures leverage state-of-the-art models such as **variational autoencoders (VAEs)**, **graph neural networks (GNNs)**, and **transformers** :

- **VAEs** offer probabilistic modeling of preferences and better handling of cold-start issues, yet are constrained by the Gaussian distribution assumption and high computational cost.
- **GNNs** are highly effective at capturing spatial and social relationships within POI-user graphs. However, they are resource-intensive (with quadratic complexity) and sensitive to missing data.
- **Transformers** excel at modeling long-range dependencies, making them well-suited for navigation sequences and textual reviews. Nevertheless, they demand substantial memory ($O(n^2)$ complexity) and require effective positional encoding schemes.

Emerging trends in recommender systems include:

- **Model hybridization** (e.g., combining VAEs with GNNs to integrate latent embeddings with structural relationships);
- **Scalability optimization** for large-scale datasets;

- **Interpretability enhancement** through techniques like Layer-wise Relevance Propagation;
- And addressing **ethical concerns**, particularly the mitigation of geographic bias that disadvantages underrepresented regions.

In conclusion, recommender systems have evolved from simple linear models to complex architectures capable of capturing non-linear interactions and rich contextual information. However, several pressing challenges persist, including scalability, fairness, interpretability, and energy efficiency. Future research must aim to develop recommendation systems that are not only smarter and more adaptive but also explainable, equitable, and sustainable.

Table 1.5 Summary of recommendation techniques

Approach	Description	Advantages	Limitations
Collaborative Filtering (CF)	Based on similarity between users or items.	Simple, effective for active users, good accuracy.	Cold start, data sparsity, poor scalability.
User-based CF	Similarity between users.	Personalized recommendations.	Requires a lot of interactions.
Item-based CF	Similarity between visited items.	More stable, less costly.	Cold start for items.
Matrix Factorization	SVD, ALS, SGD methods to extract latent factors.	Captures implicit preferences.	Popularity bias, low temporal adaptability.
Hybrid Approaches	Combines CF and content-based filtering (CBF).	Better handles cold start, more interpretable recommendations.	Complexity in integrating multiple data sources.
With Deep Learning	CNN for images, Transformers for text.	Advanced multimodal analysis.	High resource demand.
Context-aware Recommendation	Integrates time, location, weather, etc.	Dynamic, context-adapted recommendations.	Dimensional explosion, privacy concerns, real-time processing is difficult.
Advanced Approaches	Uses modern architectures: VAE, GNN, Transformers.	Complex modeling of interactions, higher relevance.	Computational cost, missing data, scalability.

Approach	Description	Advantages	Limitations
VAE (Variational Autoencoder)	Probabilistic encoder-decoder.	Handles cold start, fine-grained modeling.	Gaussian assumption, high cost.
GNN (Graph Neural Network)	Spatial/social relationships.	Rich representation of interactions.	Sensitive to graph quality.
Transformers	Models interaction sequences.	Captures long-term dependencies, very flexible.	Memory complexity, requires sequential data.
Future Trends	Model fusion, explainability, ethics, scalability.	Smarter, adaptive, and fairer recommendations.	Requires advances in AI and energy management.

1.6 Evaluation of recommender systems

Assessing the performance of a recommender system (RS) is a critical step in validating its utility, robustness, and the relevance of the suggestions it provides. A wide range of evaluation metrics has been proposed in the literature to quantify the accuracy and effectiveness of generated predictions. These indicators not only facilitate the comparison of different approaches but also help identify potential avenues for improvement [5].

This section presents the most commonly used metrics for evaluating the quality of recommendations, classified into two main categories: those assessing the accuracy of predictions (e.g., estimated ratings), and those evaluating the relevance of the recommended items. Among the most frequently employed metrics are the Root Mean Square Error (RMSE), the Mean Absolute Error (MAE), as well as information retrieval-inspired metrics such as precision, recall, and the Normalized Discounted Cumulative Gain (NDCG) [56].

The effectiveness of an RS is primarily measured by its ability to meet users' needs. The selection of evaluation metrics depends on both the nature of the processed data and the specific expectations of the target users [4]. As the field of recommender systems heavily draws from information retrieval (IR), it is common to adopt IR metrics, although certain adjustments may be necessary to suit the recommendation context.

Before proceeding with the evaluation of an RS, it is essential to distinguish between two primary modes through which recommendations are presented to users : (1) recommending the single most relevant item, and (2) recommending a ranked list of the Top-N most relevant items [57, 58]. In the first scenario, the system predicts missing ratings within the user-item preference matrix and selects the item with the highest predicted score. In the second, the system generates a ranked list of N items deemed most relevant to the user.

Evaluation largely relies on prediction accuracy, which refers to the deviation between predicted ratings and the actual ratings provided by users. Three main evaluation strategies are typically adopted for this purpose: offline evaluation, user studies, and online evaluation [57].

1.6.1 Evaluation strategies

1.6.1.1 Offline evaluation

Offline evaluation is the most widely adopted method for assessing recommender systems, primarily due to its simplicity and the absence of risk to the user experience [59]. This methodology is grounded in machine learning and relies on partitioning available data into training and testing sets to estimate the prediction error generated by the RS. A common approach involves 5-fold cross-validation, which divides the dataset into two parts: (1) a training set comprising $x\%$ of the data used to train the algorithm, and (2) a test set containing the remaining $(100-x)\%$ to evaluate the algorithm's predictions [20].

One of the key advantages of offline evaluation is that it does not require real-time interaction with users, as it simulates user behavior during system interaction [60]. As a result, it enables the testing of complex algorithms without disrupting the user experience. However, this method has certain limitations : it is insensitive to dynamic changes in user behavior and does not account for real-time factors that may affect the quality of recommendations [61]. Furthermore, some researchers argue that the quality of an RS can never be fully captured through a single evaluation method due to the multiplicity of underlying objective functions [62].

1.6.1.2 Online evaluation

Online evaluation entails real-time experimentation with actual users of the recommender system (RS). This approach involves randomly selecting a sample of users whose interactions with the system are monitored and subsequently compared with those of the remaining user base [63]. However, this methodology carries an inherent risk: exposing users to irrelevant recommendations may degrade the user experience, potentially leading to user attrition. To mitigate this risk, it is generally advised to conduct preliminary offline evaluations in order to ensure a minimum level of recommendation quality before deploying the system in a live environment [60]

A major challenge in online evaluation lies in isolating and accurately measuring the specific impact of the RS, while controlling for confounding variables. To achieve this, an A/B testing protocol is typically employed. This method contrasts a control group (A) with a test group (B), with the two differing in only a single aspect of the system's configuration [64]. For instance, in a web-based application, users in group A might be exposed to a page displaying a placebo recommendation box, while users in group B would see a page with the actual recommendation module. By comparing the performance metrics between these two groups, researchers can assess the effectiveness of the RS. Unfortunately, despite its robustness, the design and implementation of A/B testing protocols are resource-intensive and therefore infrequently applied in the evaluation of recommender systems [62].

1.6.1.3 User studies: evaluation through sampling

Sampling-based evaluation refers to a methodological approach in which a group of voluntary users is invited to interact with a recommender system (RS) while performing predefined tasks. During these experimental sessions, user interactions are systematically monitored and feedback is collected [65]. This approach enables the analysis of user behavior during interactions with the RS and facilitates the assessment of the system's impact on user decision-making. Post-interaction questionnaires provide qualitative data that complement quantitative findings, thereby offering a more comprehensive perspective on the effectiveness of the RS[66].

Nevertheless, this method poses several significant challenges. Recruiting a sufficiently large and diverse sample of participants can be difficult and may require material incentives to encourage participation. Moreover, the limited number of participants constrains the external validity of the results, making it difficult to generalize findings to broader populations [67]. Participants' time constraints also tend to limit the duration of the testing sessions, and the need to replicate scenarios to ensure result reliability increases the logistical complexity of the study[57].

Despite these limitations, evaluation by sampling provides valuable information about the user experience and actual user behavior in response to recommendations. Such insights are essential for the development of more effective and user-centered recommender systems [68].

1.6.2 Metrics used in offline evaluation

Following an in-depth analysis of the characteristics of various evaluation strategies, offline evaluation emerges as the most suitable approach within the scope of our study. Indeed, the focus of our research lies in analyzing the behavior of a large user base, making user studies less appropriate for deriving generalizable conclusions. The offline evaluation methodology enables a systematic assessment without requiring direct user interaction, thereby ensuring a reliable measurement of the recommender system's performance. In the subsequent section, we present the main metrics employed in this evaluation, highlighting both their strengths and limitations in assessing recommendation quality.

1.6.2.1 Evaluation-based metrics: predictive accuracy

The evaluation of a recommender system's performance relies on several key metrics, among which the Mean Absolute Error (MAE) and the Root Mean Squared Error (RMSE) are of primary importance. MAE computes the average of the absolute differences between predicted and actual ratings, providing an overall indication of the system's average prediction error. Lower MAE values correspond to higher predictive accuracy. In contrast, RMSE assigns greater weight to larger errors by penalizing higher deviations more strongly, thereby making it more sensitive to outliers and extreme prediction discrepancies.

a. Mean absolute error (MAE)

The Mean Absolute Error (MAE) is a common metric used to evaluate the accuracy of a model's predictions. It is the average of the absolute differences between the predicted values and the actual values [69]. The formula for MAE is as follows [70]:

$$MAE = \frac{1}{n}\sum_{i=1}^{n}(y_i - \widehat{y_i})^2 \tag{1.17}$$

Where:

- n is the number of data points,
- y_i is the actual value for the i-th data point,
- $\hat{y}_i$ is the predicted value for the i-th data point,
- $|y_i - \hat{y}_i|$ is the absolute error between the actual and predicted value.

b. Root mean squared error (RMSE)

The Root Mean Squared Error (RMSE) is a popular metric used to evaluate the accuracy of a model's predictions [71]. The formula for RMSE is as follows:

$$RMSE = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y}_i)^2} \tag{1.18}$$

1.6.2.2 Metrics based on classification

In the field of Point-of-Interest (POI) recommendation systems, decision-based metrics are commonly employed to assess the effectiveness of Top-N recommendations [??]. Originating from the domain of information retrieval, these metrics do not consider the predicted rating values generated by the system. Instead, they focus solely on the binary decision of whether or not to recommend an item [18].

The system is rewarded for including relevant items in the recommendation list and penalized for either recommending irrelevant items or failing to suggest relevant ones [19]. The primary objective is to quantify the frequency of correct versus incorrect decisions made by the recommendation engine.

Among the most widely adopted metrics are Precision, which measures the proportion of relevant items among those recommended, and Recall, which evaluates the proportion of relevant items successfully identified by the system in relation to the total number of relevant items available [25]. The F-measure offers a harmonic mean of precision and recall, thereby providing a balanced assessment of both [72]. These metrics are typically computed using a contingency table that classifies items as either relevant or irrelevant, based on their perceived importance to the user [73].

a. Precision

Precision quantifies the proportion of relevant items among the top K results returned by the system. It evaluates the system's ability to provide relevant recommendations early in the ranking. The formula used to calculate precision is presented below [19].

$$Precision@K = \frac{1}{K}\sum_{u\in U}\frac{|Rec_u \cap Test_u|}{|Rec_u|} \tag{1.19}$$

b. Recall

Recall measures the proportion of relevant items successfully retrieved within the top *K* results, relative to the total number of relevant items available in the dataset. This metric reflects the system's ability to capture the full set of relevant results and is computed using the formula shown below [19].

$$Recall@K = \frac{1}{K}\sum_{u\in U}\frac{|Rec_u \cap Test_u|}{|Test_u|} \tag{1.20}$$

c. F1-Score

The F1-score (also known as F-score or F1-measure) is the harmonic mean of precision and recall, ranging from a minimum of 0 to a maximum of 1. A high F1-score indicates that both precision and recall are simultaneously high. It treats the contributions of precision and recall as equally important. The following formula is used to calculate the F1-score [70]:

$$F1@K = \frac{2 \times Precision@K \times Recallk}{Precision@K + Recallk} \tag{1.21}$$

1.6.2.3 Ranking-based metrics

A. MAP (mean average precision)

MAP evaluates the mean precision across multiple queries by averaging the precision values at different cutoff ranks for each individual query. It captures both relevance and ranking position of results, and is particularly useful in scenarios with multiple user queries. Formula 1.22 and Formula 1.23 are employed to compute MAP [74].

$$MAP@K = \frac{1}{K}\sum_{j=1}^{M}\frac{1}{r}\sum_{k=1}^{K} Precision@k \times rel(k) \tag{1.22}$$

$$rel(k) = \begin{cases} 0, if\ POI\ at\ k_{th}\ rank\ is\ relevant \\ 1, otherwise \end{cases} \tag{1.23}$$

B. NDCG (normalized discounted cumulative gain)

NDCG is a ranking-sensitive evaluation metric that considers not only the relevance of recommended items but also their positions in the result list. It assigns greater importance to relevant items that appear higher in the ranking. This metric is especially suited for scenarios where the order of recommendations significantly impacts user satisfaction. Formulas 15 and 16 are used to compute NDCG@K [74]. Here, $\mathbf{REL_p}$ denotes the list of relevant documents (ranked by relevance) in the corpus up to position p.

$$NDCG@K = \frac{DCG@k}{IDCG@k} \tag{1.24}$$

$$DCG@k = \sum_{i=1}^{k}\frac{rel(i)}{log_2(i+1)} \tag{1.25}$$

Where IDCG is ideal discounted cumulative gain.

$$IDCG@k = \sum_{i=1}^{|REL_p|}\frac{rel(i)}{log_2(i+1)} \tag{1.26}$$

$\boldsymbol{REL_p}$ represents the list of relevant documents (ranked by order of relevance) in the corpus up to position p.

1.6.2.4 Discussion

In the evaluation of Point-of-Interest (POI) recommender systems, predictive accuracy has traditionally been regarded as the primary performance metric. However, relying solely on accuracy does not necessarily ensure an optimal user experience. User expectations vary significantly depending on the domain: for instance, recency is vital in news recommendations, whereas proximity and price are often prioritized for restaurants, and popularity may be a key factor in movie recommendations. Thus, user preferences are highly context-dependent.

Beyond accuracy, several other factors must be considered to effectively assess the quality and robustness of recommendation systems. Scalability, the system's ability to accommodate large numbers of users and items, is essential. Transparency, defined as the ability to provide understandable explanations for recommendations, fosters user trust and improves perceived system reliability. Privacy preservation is also critical to maintaining user confidence. Furthermore, aspects such as serendipity (the ability to surprise users with unexpected yet relevant items), novelty (introducing new content), and diversity (offering varied recommendations) contribute significantly to user satisfaction by enriching the overall recommendation experience.

1.7 Conclusion

This chapter has outlined the theoretical foundations of recommender systems by presenting various categories, from traditional approaches like collaborative filtering and content-based filtering to more advanced context-aware and deep learning-based techniques. Each methodology addresses specific challenges such as the cold-start problem, data sparsity, and the integration of contextual user information.

Special emphasis was placed on performance evaluation, a critical step for validating the relevance and effectiveness of generated recommendations. Evaluation strategies and metrics such as RMSE, precision, recall, and NDCG enable a comprehensive comparison of different methods, guiding methodological decisions in alignment with system objectives.

Collectively, these elements provide a strong foundation for understanding the current challenges in recommender systems and pave the way for the introduction of adaptive and intelligent solutions in subsequent chapters.

Chaptre 2 POI recommendation in LBSNs

2.1 Introduction

The recommendation of Points of Interest (POIs) within Location-Based Social Networks (LBSNs) presents a complex and intriguing challenge. LBSNs provide a wealth of contextual, social, and geographic data, which, when properly leveraged, can significantly enhance the accuracy and relevance of recommendations.

This chapter explores various aspects of POI recommendation within LBSNs, starting with an overview of POIs and LBSNs. It defines their characteristics and analyzes the data generated around POIs. We will also address the key challenges associated with POI recommendation, such as data sparsity, the heterogeneity of POIs, and the cold-start problem. Finally, we will review modern recommendation techniques, including collaborative filtering, graph neural networks, and multimodal models, to illustrate the solutions proposed in the literature to overcome these challenges. This chapter highlights recent advancements and future prospects in the field, while identifying persistent gaps that require new approaches for more precise and contextually relevant recommendations.

2.2 POIs and LBSNs

POIs, such as restaurants, museums, and parks, are fundamental elements of Location-Based Social Networks (LBSNs) like Foursquare, Yelp, or Instagram [75]. These platforms enable users to share geolocated data, thus providing a rich source for analyzing human behavior and generating personalized recommendations.

At the intersection of geolocation, social interactions, and individual preferences, POIs power intelligent recommendation systems and contribute to modeling mobility dynamics. Their study lies within an interdisciplinary research field that combines human geography, network analysis, and machine learning.

2.2.1 Definition of POIs (Points of Interest):

POIs are geographic locations deemed interesting to users. This can include:

- Tourist attractions (museums, monuments),
- Public infrastructures (train stations, hospitals),
- Commercial establishments (restaurants, cafés, hotels),
- Natural spaces (beaches, parks).

2.2.2 Definition of LBSNs (Location-Based Social Networks):

LBSNs are social networks that leverage users' location data to enrich their interactions. Popular examples include Foursquare, Yelp, Swarm, and Google Maps (social layer) [76].

These platforms enable users to:

- Share their location through check-ins,
- Post reviews, ratings, photos, etc.,
- Interact with other users (comments, likes, recommendations).

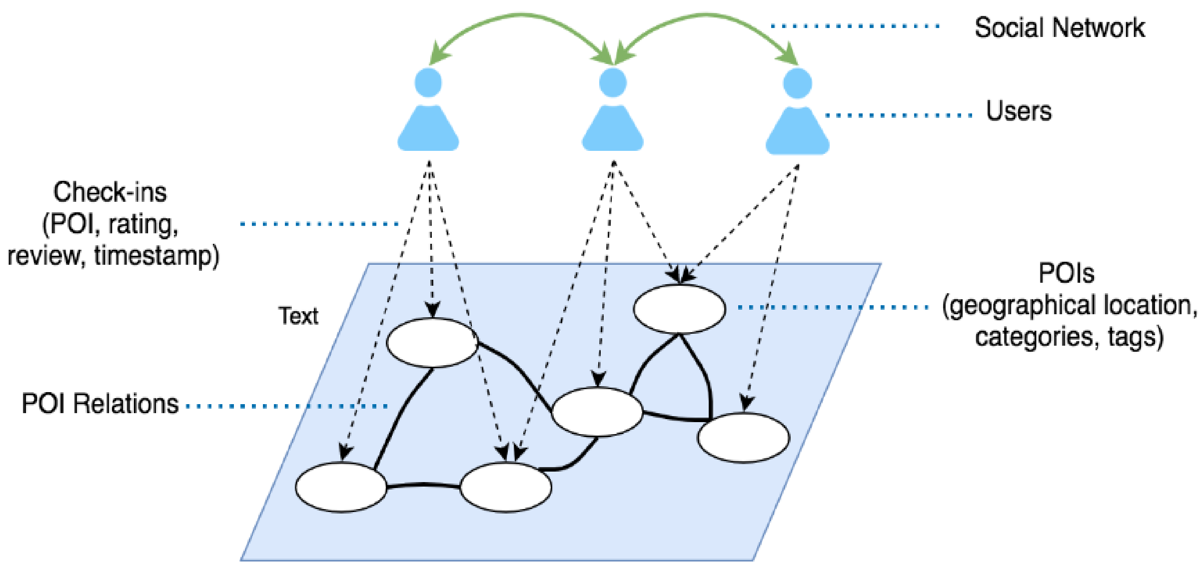

Figure 2.1 A Location-based Social Network [34]

2.2.3 Characteristics of POIs in LBSNs

2.2.3.1 Spatial and Temporal Attributes

POIs are characterized by spatial attributes (GPS location, urban or rural context) and temporal attributes, which influence their visitation patterns. Spatially, the exact location of a POI determines its accessibility and reach, while the density (urban vs rural areas) modulates its visibility. Temporally, dynamics such as seasonality (e.g., increased summer attendance at parks) or opening hours (e.g., peak hours at Museums during the night) reveal key behavioral patterns for analysis and recommendation [77].

For example, POIs incorporate spatial dimensions (GPS, urban/rural context) and temporal dimensions (seasonality, hours of operation), whose analysis allows for the anticipation of user behavior (e.g., park visitation in summer or café attendance at night).

2.2.3.2 Social and Contextual Attributes

POIs are also defined by social and contextual dimensions that shape their attractiveness. Their popularity can be measured through quantitative indicators, such as the number of check-ins or reviews left by users [24]. Furthermore, social relationships play a crucial role: recommendations from friends (e.g., "My friends visit this café") significantly influence individual choices. Additionally, the situational context such as weather conditions, local events (e.g., a festival), or the mode of transportation used modulates the relevance of a POI at a given moment [78]. These attributes enhance behavioral analysis and optimize recommendation systems by integrating human and environmental dynamics.

Indeed, the attractiveness of POIs depends on social factors (popularity, peer influence) and contextual factors (weather, events, transportation). These elements, combined with geo-temporal data, allow for more refined prediction of user preferences.

2.2.3.3 Multimodal Data

POIs rely on multimodal data that enrich their representation and enhance their discoverability. Textual content such as descriptions, tags (e.g., veganrestaurant), or comments captures qualitative and thematic information. Images shared by users provide a concrete visualization of the place, increasing engagement and trust. Finally, structural metadata (categories like "art museum," price range, or accessibility indicators) enable precise filtering and optimal integration into recommendation engines [79]. This diversity of data, combined with multimodal AI techniques (natural language processing, computer vision), enables a richer and more personalized modeling of POIs.

2.2.3.4 The Role of POIs in LBSNs

POIs lie at the heart of LBSN dynamics. They serve as spatial anchors for:

- Collecting users' mobility behaviors,
- Analyzing personal and community preferences,
- Providing personalized recommendations for places to visit,
- Studying social and geographical patterns.

2.2.3.5 Data Generated Around POIs in LBSNs

Each POI is associated with various types of data relevant to recommendation systems:

- Check-ins (visits with timestamps and location),
- User ratings and reviews,
- Shared photos and hashtags,
- Categories of the location (restaurant, Mosque, cinema, etc.),
- Geographical proximity to other POIs,
- Social relationships of visitors (friends, followers).

Discussion

POIs play a central role in geosocial applications, as evidenced by recommendation systems such as those used by Foursquare (based on check-ins and social history), contextual integration in Google Maps (adjustments based on time or traffic), and Yelp's multi-criteria approach (combining reviews, photos, and metadata). These implementations allow for the creation of precise mobility profiles and the offering of contextualized suggestions (e.g., location, time, weather), while contributing to the study of spatial social networks and urban dynamics.

Current research challenges suggest innovative directions: mitigating spatial biases (such as the overrepresentation of affluent neighborhoods), modeling dynamic POIs (e.g., food trucks, temporary events), and maintaining a delicate balance between data anonymization and analytical utility. The emergence of generative AI (via Large Language Models, LLMs) also promises significant advancements, such as the automatic generation of personalized POI descriptions. These prospects highlight the need for hybrid approaches, combining implicit trust models, co-visit analysis, and privacy protection, to build systems that are both intelligent and ethical.

2.3 Challenges in POI Recommendation in LBSNs

Point of Interest (POI) recommendation systems in Location-Based Social Networks (LBSNs) (e.g., Foursquare, Yelp) face unique challenges due to the dynamic nature of spatial, social, and contextual data. These systems must not only predict relevant locations but also integrate factors such as human mobility, changing preferences, and geographical constraints.

In the following section, we analyze the major challenges and proposed solutions discussed in the literature:

2.3.1 Data Sparsity

One of the major issues is the extreme sparsity of user-POI matrices. In most cases, a user interacts with only a small fraction of the available POIs, resulting in a large number of missing values, which complicates the learning of reliable models [80]. Some of the solutions proposed in the literature include:

- **Contextual Enrichment**: Integrating external data such as weather, opening hours, or local events increases the informational density and helps explain certain visit behaviors.
- **Transfer Learning** : This technique involves transferring knowledge acquired from similar locations (in terms of type or environment) to predict preferences for new POIs.

Example :

On Foursquare, the average user visits fewer than 100 POIs out of millions of available options.

Problem:

The user-POI matrices are extremely sparse (95-99% of missing values), as users typically interact with only a fraction of the available POIs.

Proposed Solutions :

1. **Contextual Enrichment**: By integrating external data (e.g., weather, events, opening hours), this approach helps explain visit patterns. For example, a user may choose to visit a café during rainy weather, with the weather data helping to justify this choice.
2. **Transfer Learning** : This solution transfers knowledge from a "source" POI (e.g., a popular Italian restaurant) to a "target" POI that is similar but less frequently visited, using shared embeddings between POIs of the same category.

2.3.2 Heterogeneity of Points of Interest (POIs)

Points of Interest (POIs) exhibit significant variability in terms of size, function, location, and popularity. For instance, a small neighborhood café differs in both characteristics and visitation frequency compared to a national museum [81]. To address this variability, we propose the following approaches:

- **Hierarchical Clustering**, which organizes POIs into homogeneous groups based on structural characteristics such as geographical proximity, category, and ambiance.
- **Vector Representations (Embeddings)**, which utilize deep learning techniques, particularly Graph Neural Networks (GNN), to capture rich and semantic representations of POIs by considering their attributes and topological relationships.

Example :

A national museum attracts international visitors, while a local café primarily serves neighborhood patrons.

Problem:

POIs vary in size, popularity, and function (e.g., a café versus a museum), making uniform modeling challenging.

Solutions :

1. **Hierarchical Clustering** can group POIs based on similarities (e.g., category, surrounding population density) using algorithms such as k-means on geographical and social features.
2. **Graph Neural Networks (GNN)** can capture relationships between POIs through heterogeneous graphs (e.g., co-visits, proximity). The POI-GCN model propagates preferences among neighboring POIs.

2.3.3 Selection Bias

Data from Location-Based Social Networks (LBSNs) are often biased towards more popular locations, such as famous landmarks or restaurant chains. This bias skews recommendations, marginalizing less frequented but potentially interesting locations

[82]. To mitigate this issue, we recommend the following strategies:

- **Reweighting**, which adjusts the weights assigned to interactions based on the frequency of POI appearances to counteract the overrepresentation of popular places.
- **Result Diversification**, which focuses on developing algorithms that promote the exploration of alternative or "niche" POIs to enhance recommendation diversity and encourage discovery.

Example :

The Eiffel Tower receives 1,000 check-ins per day, while a local art gallery might receive only 2.

Problem :

Popular POIs (e.g., monuments) dominate the data, sidelining relevant "niche" locations.

Solutions :

1. **Reweighting interactions** by assigning inverse weights based on POI frequency (e.g., Inverse Frequency Weighting).
2. **Maximizing recommendation diversity** using metrics like entropy or geographical dispersion, and employing methods such as Determinantal Point Processes (DPP) to promote variety.

2.3.4 Cold Start Problem

Two main forms of the cold start problem arise in Location-Based Social Networks (LBSN) [83]:

- **Newly Added Points of Interest (POIs) :** These POIs lack any check-in history. In such cases, metadata (e.g., categories, geographic coordinates, images) can be leveraged to estimate their potential attractiveness.

- **New Users:** Users without any prior activity, which makes personalized recommendations challenging. Solutions involve utilizing social profiles (e.g., friends' interests, group memberships, habitual locations) or demographic data.

Example :

1. For newly added POIs (without any visit history), metadata (e.g., category, images) and geographical similarities can be utilized.
2. For new users (without check-in history), social profiling (e.g., friends, interests) or Zero-shot learning can be employed to infer preferences from textual data.

2.3.5 Contextual and Temporal Challenges

A major challenge in Point of Interest (POI)-based recommendation systems lies in their heavy reliance on contextual factors (e.g., time of day, weather, company) and the temporal evolution of user preferences [31]. For instance, an individual may prefer parks during the day on weekends but opt for café in the evening on weekdays. This variability requires models that can dynamically adapt to changing contexts.

Several approaches have emerged to address this challenge :

- **Sequential Models (Transformers, RNNs):** These architectures capture visit sequences to predict the next POI, incorporating temporal patterns (e.g., weekly habits).
- **Reinforcement Learning (e.g., Deep Q-Learning):** This method allows real-time adjustments to recommendations based on user feedback, thereby optimizing contextual relevance.

However, these methods struggle to generalize in the face of the complexity of human behaviors (e.g., occasional exceptions) and data latency (e.g., delays in context updates).

Example :

- A user visits parks on weekends and cafés in the evening.
- **Problem:** Preferences depend on context (e.g., time, weather, company) and evolve over time.
- **Solutions:** Sequential models (e.g., Transformers or RNNs) can predict the next POI based on historical data, while reinforcement learning techniques (e.g., Deep Q-Learning) enable real-time adaptation of recommendations based on user feedback.

2.3.6 Synthesis

Location-based social networks (LBSN) recommendation systems for Points of Interest (POI) require an integrated, multidimensional approach to address the specifics of data and user behaviors in mobile contexts. For instance, managing data sparsity relies on contextual enrichment through external information such as weather or local events, as well as transfer learning techniques that leverage knowledge from similar POIs. The modeling of the heterogeneity of locations, which vary widely in terms of size, category, or attractiveness, is made possible through hierarchical clustering methods or learning embeddings via Graph Neural Networks (GNN), which capture complex structural relationships.

Simultaneously, reducing selection bias, particularly in favor of the most popular locations, necessitates reweighting and diversification mechanisms. These are designed to encourage a more equitable exploration of POIs, including less-visited ones. The cold start problem, whether it concerns new places or new users, is addressed by utilizing metadata (such as category, location, and image) for the former and social or demographic profiling for the latter.

Recent advancements in artificial intelligence open new possibilities. Language models (LLMs), for instance, allow for the automatic generation of place descriptions or the prediction of preferences from contextual prompts. On the other hand, Graph Neural Networks (GNN) provide powerful tools for analyzing co-visit networks, uncovering latent similarities between users and POIs.

However, these advances also raise critical issues. Privacy remains a major concern, particularly regarding the anonymization of geolocation data, which must ensure both user confidentiality and the relevance of the analyses. Moreover, spatial fairness must be considered, as some algorithms may inadvertently reinforce territorial inequalities, such as the marginalization of less-connected rural areas.

In conclusion, although LBSN systems today possess powerful tools for delivering hyper-personalized recommendations, their responsible use requires interdisciplinary collaboration, combining expertise in AI, urban geography, ethics, and digital sociology. Future research should incorporate mechanisms for algorithmic transparency (model explainability) and equitable representation of POIs, while harnessing the potential of neuromimetic architectures such as spatiotemporal Transformers for increasingly fine-grained and inclusive personalization.

2.4 POI Recommendation Techniques in LBSNs

Points of Interest (POIs) are central elements in Location-Based Social Networks (LBSNs) such as Foursquare, Yelp, or Google Maps. POI recommendation aims to suggest relevant locations based on user preferences, spatiotemporal context, and social interactions. This section explores the primary techniques used to model these systems, focusing on their mechanisms, advantages, and limitations.

2.4.1 Context-based Collaborative Filtering

Traditional collaborative filtering relies on past interactions between users and POIs. In the context of an LBSN, this method is enhanced by integrating contextual variables such as time, weather, or day of the week. This allows for personalized recommendations based on the user's current situation or time [24]. For instance, if a user regularly checks in at a café in the morning, the system will recommend nearby cafés at that time. Some of the methods used in Contextualized Collaborative Filtering include:

- **User-POI Matrix with Context**: This is a tensor representation: $R \in Rm \times n \times c$, where c represents the context (e.g., time, weather) [84]. For example, if a user frequently visits a café near their workplace at noon, the system will recommend this café when the user is nearby and it is noon.
- **Tensor Factorization**: This involves latent factor decomposition to reduce dimensionality while incorporating the context (such as Tucker Decomposition or CP Decomposition) [85].

2.4.2 Graph Neural Networks (GNNs)

Graph Neural Networks (GNNs) handle data in the form of heterogeneous graphs, where nodes can represent users, POIs, or categories, and edges represent relationships such as "check-in," "friendship," or "geographical proximity." GNNs learn to propagate preferences and information across these relationships [86]. For example, if a user's friend frequently visits a nearby park, the likelihood of recommending that park increases due to information propagation within the graph.

2.4.3 Transformers for Visit Sequences

Transformers, known for their ability to handle sequences, are applied here to analyze the temporal order of check-ins. Through multi-head attention mechanisms, they identify which previous visits most influence the user's next destination [87]. This allows for the modeling of sequential and temporal behavior in users. For instance, if a user frequently visits a restaurant followed by a cinema, the model can predict a cinema as the next POI after a restaurant visit [88].

2.4.4 Multimodal Fusion

Data derived from Location-Based Social Networks (LBSN) is inherently multimodal, encompassing text, images, and metadata. Multimodal fusion techniques combine these diverse sources to enhance the quality of recommendations. For example, recommending a restaurant based on user-uploaded photos and associated reviews. This technique integrates multiple types of data: textual (reviews, descriptions), visual (photos associated with locations), and spatial (GPS coordinates). These data are incorporated into a unified model, often through deep neural networks, to enable a rich and diverse understanding of user preferences [89]. To achieve this, architectures such as BERT can be used to encode textual descriptions, while CNNs are employed for images, with the embeddings being concatenated to form a unified representation.

2.4.5 Discussion

Recommendation systems for Points of Interest (POI) in Location-Based Social Networks (LBSN) have significantly evolved, transitioning from traditional approaches like collaborative filtering to more advanced models such as Graph Neural Networks (GNN) and Transformers. Each method presents its strengths and weaknesses in terms of accuracy, interpretability, and computational cost.

Contextualized collaborative filtering allows for dynamic adaptation to changes in user behavior but suffers from scalability issues and a bias towards popular POIs. Graph-based models (GNN) excel in modeling spatial and social relationships, but their computational complexity ($O(n^2)$) makes them difficult to apply at scale. On the other hand, Transformers efficiently capture long-term dependencies in visit sequences, but their high memory cost and reliance on positional encoding limit their use in certain cases.

Hybrid methods (such as VAE + GNN or Transformers + GNN) offer a promising solution by combining the strengths of multiple approaches, but at the cost of increased complexity and significant resource requirements. In the future, the integration of generative AI and federated learning could help overcome some of the current limitations, particularly concerning bias and privacy concerns.

Table 2.1 below analyzes the trade-offs between performance, complexity, and applicability of the various methods described earlier.

Table 2.1 Advantages and disadvantages of recommendation methods

Method	**Advantages**	**Disadvantages**
Contextualized Collaborative Filtering	Captures dynamic preferences Models contextual dependencies	High algorithmic complexity Bias towards popular POIs Sensitive to missing data
GNN (Graph Neural Networks)	Effectively models spatial/social relationships Good interpretability.	High computational cost ($O(n^2)$) Sensitive to incomplete graphs
Transformers (Sequences)	Handles long dependencies without memory loss Adaptable to multimodal data.	High memory usage ($O(n^2)$ for attention) Requires positional encoding
Hybrid Methods (VAE+GNN, Transformers+GNN)	Robust to cold start. Flexible for heterogeneous data.	Implementation complexity. Very high computational cost.

2.5 State of the Art of POI Recommendation in LBSNs

2.5.1 Modeling Geographical Influences from Check-ins

In the context of smart tourism, user profiles are defined as a rich set of characteristics encompassing travel preferences (such as favorite destinations, duration of stays, preferred seasons, available budget, etc.), interests (e.g., hiking, museums, shopping, gastronomy), and travel history (including visited locations, activities engaged in, types of accommodations, transportation used, etc.). These profiles are generally constructed from digital data collected through various technologies, including mobile applications, websites, geolocated social networks (LBSNs), and Internet of Things (IoT) devices

[90]. Collaborative filtering (CF) techniques utilize these profiles to recommend Points of Interest (POIs), relying on the behaviors and preferences of similar users. In this context, Ye et al. [91] proposed a geo-social CF model that combines geographical influences and social connections, integrating the proximity of POIs with user check-ins. Their approach demonstrated a significant improvement in recommendation accuracy compared to traditional CF models.

Additionally, Cheng et al. [92] introduced a matrix factorization method that incorporates spatial distance functions to better represent users' preferences in relation to the geographical dimension. This method also outperformed basic CF approaches.

Similarly, Lian et al. [93] developed the GeoMF model, which merges matrix factorization with a geographical model. By exploiting both geographic distance and check-in data, GeoMF enhances recommendation accuracy, surpassing several existing geographical models.

Finally, Zhang et al. [94] presented the Geosoca model, which simultaneously leverages both social and geographical correlations, considering the proximity of POIs and their categories within the recommendation process. This model proved its effectiveness by outperforming traditional CF-based geographical approaches.

Table 2.2 CF work using check-ins to model geographical influences

Authors	**Model / Method**	**Principle**	**Data Used**	**Advantages / Contributions**
Ye et al.	Geo-social CF	Combines geographical and social influences	Check-ins, geographical proximity, social links	Improves accuracy compared to traditional CF
Cheng et al.	Matrix Factorization + Spatial Distance	Integrates spatial distance functions into factorization	Check-ins, geographical distances	Better represents geo-dependent preferences
Lian et al.	GeoMF	Combines matrix factorization and geographical modeling	Check-ins, geographical distances	Outperforms several geographical models in accuracy

Authors	**Model / Method**	**Principle**	**Data Used**	**Advantages / Contributions**
Zhang et al.	Geosoca	Integrates proximity of POIs and categories to capture spatio-social correlations	Proximity of POIs, categories, social networks	Superior performance compared to traditional geographical CF models

2.5.2 Spatio-Temporal Modeling from Contextual Check-ins

In contrast to traditional CF approaches, which solely rely on geographic influences to recommend Points of Interest (POIs) (Table 2.2), Liu et al. [95] proposed an innovative model that leverages check-ins while incorporating contextual factors such as time and user profile. By utilizing machine learning algorithms, this model effectively captures user preferences, leading to a significant improvement in recommendation accuracy compared to traditional CF models.

Similarly, Zhao et al. [96] developed Geo-Teaser, a model that combines geo-temporal sequential embedding with ranking techniques. This model exploits geo-temporal check-in sequences to better capture user preferences, outperforming methods that focus solely on geographical or temporal aspects.

On the other hand, Wang et al. introduced ST-RNN, a spatio-temporal recurrent neural network. This model enables the modeling of spatial and temporal dependencies between check-ins, addressing missing data issues and enhancing the quality of POI recommendations [97];

In contrast, Lian et al. [10] opted to exclude social and contextual data, focusing exclusively on check-ins. They proposed LightRec, a recommendation system optimized for handling large volumes of data, capable of delivering high-quality recommendations while maintaining low computational complexity.

Finally, Guo et al. [34] adopted a graph-based approach, integrating check-ins, reviews, and social relationships. Their model captures the complex interactions between users and POIs, significantly improving recommendation accuracy.

Table 2.3 below summarizes POI recommendation work utilizing check-ins enriched with contextual, sequential, or structural data :

Table 2.3 Contextual approaches based on check-ins for spatio-temporal analysis

Author(s)	**Proposed Model**	**Approach**	**Key Features**	**Improvements Made**
Liu et al.	Contextual CF	Machine learning on check-ins + context (time, user category)	Integration of contextual data into user preference modeling	Higher accuracy compared to traditional CF models
Zhao et al.	Geo-Teaser	Geo-temporal sequential embedding + ranking	Analysis of geo-temporal check-in sequences to capture dynamic preferences	Better accuracy vs. geo- or time-based models alone
Wang et al.	ST-RNN	Spatio-temporal RNN on check-ins	Modeling spatial and temporal dependencies using recurrent neural networks	Solved sparsity problem, improved recommendations
Lian et al.	LightRec	Light recommendation based solely on check-ins	Avoids social/contextual data; efficient processing of large databases	Good accuracy with low computational cost
Guo et al.	Graph-based Model	Exploitation of user-POI graphs (check-ins, reviews, relations)	Modeling complex interactions using graph structures	Richer and more precise recommendations through relational perspective

2.5.3 Modeling the Spatio-Temporal and Social Context

Several recent studies have focused on enhancing the contextual richness of point-of-interest (POI) recommendations by simultaneously integrating spatial, temporal, social, and behavioral factors. Chen et al. [98] proposed an approach that combines spatial and temporal dimensions, enabling the capture of user dynamics over time and space. This model improves the accuracy of targeted recommendations based on the time of day or location.

Similarly, Ding, Chen, and Li [99] introduced an innovative spatio-temporal distance metric that considers both the geographic proximity of POIs and the users' specific temporal preferences. This method led to a significant improvement in performance for temporal recommendations.

Gao et al. [100] developed a pairwise ranking method leveraging geo-social correlations. Their approach integrates spatial proximity and social relationships among users, enhancing the quality of rankings by combining individual preferences with social influence.

Moreover, notable progress has been made in integrating textual and temporal data. Huang et al. [101] designed the STPR model, which utilizes spatio-temporal effects and introduces a personalized ranking system based on user intentions. This model has demonstrated clear superiority in predicting the next POI to be visited.

In real-time recommendation, Jiao et al. [102] proposed R2SIGTP, an adaptive system that considers users' evolving geographic and temporal preferences. This system provides instant and contextually relevant POI recommendations.

Wang et al. [103]proposes a trust-enhanced collaborative filtering (TCF) approach for personalized Point-of-Interest (POI) recommendation. By integrating trust relationships into the traditional collaborative filtering framework with geographic influences and temporal influence for POI recommendation, this model outperforms traditional approaches in terms of precision and recall.

Finally, Zhang et al. [104] introduced a personalized geographic influence model that integrates not only the distance between locations but also each user's specific geographic preferences, enabling further personalization of recommendations.

Table 2.4 presents a comparative summary of the major contributions in this area, highlighting the techniques used, the types of data leveraged, and the benefits brought by each method.

Table 2.4 Spatio-Temporal and Geo-Social Approaches for POI RS in LBSNs

Author(s)	**Model/Approach**	**Integrated Factors**	**Main Contribution**
Chen et al. [98]	Spatio-temporal model	Spatial, temporal	Capturing user behavior over time and space to refine recommendations.
Ding, Chen et Li [99]	Integrated spatio-temporal distance	Spatial, temporal	Improving recommendations by considering geographical proximity and temporal preferences.
Gao et al. [100]	Pairwise ranking based on geo-social correlations	Spatial, social	Better ranking of POIs through the integration of social relationships and geolocation.
Huang et al. [101]	STPR (Spatio-Temporal and Preference Ranking)	Spatial, temporal, user intention	Personalized prediction of the next POI to visit, surpassing classical methods.
Jiao et al. [102]	R2SIGTP (Real-time Recommendation)	Spatial, temporal	A dynamic system that adapts to changing user preferences.
Wang et al. [103]	Contextual POI model	Trust, geographic influence and temporal influence	Finer recommendations based on the specific contexts of each location.
Zhang et al. [104]	Personalized geographical influence model	Geographical distance, user preferences	Integration of geographical preferences unique to each user.

2.5.4 Discussions:

Previous studies clearly illustrate the evolution of point-of-interest (POI) recommendation techniques within Location-Based Social Networks (LBSNs), with a clear shift towards advanced models based on deep learning and graph-based architectures. While earlier approaches primarily focused on check-in data, recent research has expanded the scope by integrating the semantics of geographic correlations, as well as encoder-based architectures.

Among the most representative models, BayMAN (Bayes-enhanced Multi-view Attention Networks) stands out for its simultaneous use of semantics, geographic distance, and individual preferences. By constructing multi-view graphs of POIs, this model enhances the understanding of complex relationships between places while mitigating the impact of unreliable check-in data.

In a similar vein, MERec (Meta-learning Enhanced POI Recommendation) introduces a dual-encoder mechanism, combining a category-level encoder for common user behaviors and a POI-level encoder to capture transitions specific to urban journeys. This strategy strengthens the model's predictive ability for the next recommended location.

Continuing in this direction, the TARE (Event-Based Probabilistic Embedding) model, proposed by Zhang et al., focuses on capturing geographic influences through check-in activities localized within specific regions. By jointly integrating temporal, geographic, and semantic factors, this approach demonstrates strong performance compared to existing state-of-the-art models.

These advancements confirm the importance of considering a variety of contextual factors spatial, temporal, social, and textual to improve the relevance and accuracy of recommendations. However, it is noteworthy that, to date, no approach seems to fully leverage the combination of POI visitation and the similarity of user trajectories. This gap represents a promising research direction, particularly in urban environments where movement trajectories reveal rich implicit preferences.

2.6 Conclusion

In conclusion, this chapter highlighted the various challenges and techniques associated with POI recommendation in Location-Based Social Networks (LBSNs), particularly the issues of data sparsity, POI heterogeneity, and the contextual and temporal factors involved. While existing solutions are effective in certain contexts, they still struggle to provide fully context-aware and relevant recommendations, particularly when it comes to simultaneously considering the geographic and temporal characteristics of users.

In this regard, the thesis presents an innovative contribution in the following chapter by introducing a novel similarity measure that explicitly incorporates the spatiotemporal context of users and the similarity of their visit trajectories. This approach allows for a more accurate capture of user interactions within LBSNs, ultimately enhancing the precision of POI recommendations.

PART TWO

CONTRIBUTION

Chaptre 3 The SPPUR model

3.1 Introduction

This chapter presents our SPPUR (Similarity of Paths and the Proximity of Users for Recommending POIs) model, a new approach to POI recommendation that integrates both historical check-ins and user journeys in location-based social networks (LBSNs) [105]. While traditional recommendation systems focus mainly on individual check-ins, our model also exploits visit sequences between POIs to better capture actual tourist behavior. We first formulate the problem by highlighting the limitations of existing approaches, notably their inability to account for complete user journeys and geographical proximities. Next, we detail the SPPUR-based recommendation process, illustrated by a concrete example, before formally describing our model, which combines an adaptation of the TF-IDF technique with a geographical similarity measure. Experimental validation, carried out on real Foursquare data, compares three variants of SPPUR with each other and with other conventional similarity measures, using standard metrics (PRECISION, RAPPEL, MAP, NDCG).

3.2 Problem definition

The rapid urbanization of cities has led to an explosion in the number of points of interest (POIs), including restaurants, hotels, museums and other places frequented by tourists. At the same time, the intensive use of location-based social networks (LBSNs), such as Geolife, Facebook, Gowalla, or Foursquare, is capturing in detail user behavior and their current and future needs. In this context, POIs' recommendation systems (RS) play a crucial role in offering personalized services to new tourists on their travels.

Among recommendation methods, user-centered collaborative filtering (CF) is widely used due to its simplicity and effectiveness. This approach is based on two key steps:

- The selection of users most similar to the active user, based on their past interactions with POIs.

- Predicting check-ins for unvisited POIs, by exploiting the preferences of similar users.

However, the quality of FC-generated recommendations is highly dependent on the similarity measure used to assess relationships between users. A poorly adapted measure can lead to inaccurate predictions, negatively impacting the user experience. Although several similarity measures (e.g. cosine, Pearson) are commonly used in the literature, few of them incorporate essential dimensions such as path similarity and geographical closeness between users and POIs.

Moreover, POI recommendation systems in LBSNs face major challenges:

- **Data sparsity**: Users visit only a tiny fraction of available POIs, making user-POI matrices extremely hollow.

- **Cold start**: New users or POIs lack interaction history, limiting the ability of models to provide relevant recommendations.

To answer this question: How can we design a similarity measure capable of capturing both behavioral and geographic similarities between users, in order to provide accurate POI recommendations that are robust to the problems of data scarcity and cold start? We have designed a new similarity measure that integrates two key dimensions:

- The similarity of user paths, taking into account visit sequences.

- Geographical similarity between users and their POIs of departure and arrival.

3.3 Problem Formulation

To address the issue outlined in Section 3.1, we propose a novel similarity measure named SPPUR (Similarity of Paths and the Proximity of Users for Recommending

POIs), inspired by the TF-IDF technique [106]. This measure is specifically designed to capture both the similarity of user trajectories and their geographic proximity to visited POIs, leveraging check-in data and the sequential order of visits.

To present our approach in a clear and structured manner, we divide the explanation into three main steps:

- Definition of similarity computation formulas: We introduce the equations used to assess relationships between users, taking into account their travel patterns and geographical proximity.
- Prediction based on similarity scores: Once the similarity values are computed, we describe how they are used to predict potential check-ins for active users.
- Practical illustration: Through a concrete example, we demonstrate how the SPPUR similarity matrix is derived from the User-POI-check-in matrix, by applying the previously defined formulas.

3.3.1 Recommendation Process Based on the SPPUR Similarity

In this subsection, we compute the SPPUR similarity using three types of similarity measures. The first type accounts for user preferences based on their POI check-ins observed along their trajectories. The second and third types focus exclusively on users' choices of departure and arrival points. These three similarity components are then combined to form the final SPPUR similarity measure. This composite similarity is subsequently used to compute predictions that support the POI recommendation process.

3.3.1.1 User–User Similarity Based on POI Paths

In the following, we focus on user profiles inferred from tourist check-ins and their corresponding POI trajectories. We hypothesize that the similarity between users can be deduced from the resemblance between the sequences of POIs visited during their trips. Each user profile is thus represented as a sequence of character strings encoding the ordered list of POIs visited.

Accordingly, the similarity between two users can be computed by adapting the **TF-IDF** (Term Frequency–Inverse Document Frequency) technique [21, 107]to the domain of smart tourism. In this context : (1) each term is mapped to a POI, and (2) each trajectory of visited POIs is treated as a document. This adaptation enables us to evaluate the importance of individual POIs (terms) within the travel paths (documents) of various users.

To this end, we first compute the frequency of each POI within a user's profile (Formula 3.1). Next, we assess the global significance of each POI by measuring the number of users who have visited it (Formula 3.2). We then derive the weighted score for each (User, POI) pair using Formula (3.3). Finally, based on these individual scores, we estimate the similarity between each pair of users using Formula (3.4), as described below.

A) TF Calculation

To compute the **Term Frequency (TF)** value for a POI POIi in the profile of a given user Usera, we rely on the number of times the user has checked into this POI. The calculation is defined as shown in Formula (3.1) below:

$$TF(User_a, POI_i) = \frac{Freq(User_a, POI_i)}{Visits(User_a)} \quad (3.1)$$

With :

- $\mathrm{Freq}(\mathrm{User_a}, \mathrm{POI_i})$ = Number of check-ins of $\boldsymbol{User_a}$ on $\boldsymbol{POI_i}$

- $Visits(User_a)$ = Total check-ins of $\boldsymbol{User_a}$ on all POIs.

B) IDF Calculation

To assess the global importance of a given POIi, we compute its **Inverse Document Frequency (IDF)** using the logarithm of the ratio between the total number of users and the number of users who have visited POIi, as shown in Formula (3.2) below:

$$IDF(POI_i) = log \times \frac{NUsers}{POI_i(users)} \tag{3.2}$$

With:

- NUsers = Total number of Users
- $POI_i(users)$ = Number of Users who visited POI_i

C) TF-IDF Calculation

To determine the relevance of a POI_i for a given user $User_a$ relative to all POIs and users, we compute the **TF-IDF score** for each ($User_a$, POI_i) pair using Formula (3.3) below :

$$TFIDF(User_a, POI_i) = TF(User_a, POI_i) \times IDF(POI_i) \tag{3.3}$$

D) Sim_{path} Similarity Calculation

The TF-IDF scores computed in Formula (3.3) are then used to define the **trajectory-based similarity** denoted as $\boldsymbol{Sim_{path}}$ between each pair of users. This similarity is calculated according to Formula (3.4) below:

$$Sim_{path}(user_a, user_b) = \frac{\sum_{i\in I}\left(TFIDF(user_a, POI_i) \times TFIDF(user_b, POI_i)\right)}{\sqrt{\sum_{i\in I}\left(TFIDF(user_a, POI_i)\right)^2} \times \sqrt{\sum_{i\in I}\left(TFIDF(user_b, POI_i)\right)^2}} \tag{3.4}$$

Where:

– I represent the set of $POIs$ visited by $user_a$ and $user_b$.

3.3.1.2 User–User Similarity Based on Starting POI

We posit that two users Usera and Userb are similar if their respective first visited POIs are geographically close. Accordingly, their **departure-based similarity**, denoted as Simstart , can be estimated using Formula (3.5) below:

$$Sim_{start}(user_a, user_b) = \frac{1}{1 + dist_{start}(user_a, user_b)} \tag{3.5}$$

Where:

- $\boldsymbol{dist_{start}}(user_a, user_b)$ represent the distance between the two users $\boldsymbol{user_a}$ et $\boldsymbol{user_b}$ based on their initial locations, which depend on the first POIs they visited.

3.3.1.3 User–User Similarity Based on Final POI

Similarly, we assume that two users a and b are similar if their last visited POIs are located near each other. The **arrival-based similarity**, denoted as Simend , is computed using Formula (3.6) below:

$$Sim_{end}(user_a, user_b) = \frac{1}{1 + dist_{end}(user_a, user_b)} \tag{3.6}$$

Where:

- $\boldsymbol{dist_{end}}(user_a, user_b)$ represent the distance between the two users $\boldsymbol{user_a}$ et $\boldsymbol{user_b}$ based on their final locations, which depend on their last visited POIs.

To compute the distance between two POIs, we used the **Haversine formula**, a widely used method in navigation and geographic information systems [108] [109].

$$dis = 2r \cdot arcsin\sqrt{sin^2\left(\frac{\Delta lat}{2}\right) + cos(lat_1) \cdot cos(lat_2) \cdot sin^2\left(\frac{\Delta long}{2}\right)} \tag{3.7}$$

With:

- dis is the distance betweentwoPOIs(km),
- r is the earth's radius (6,371 (km)),
- $lat_1$and lat_2 are the latitudes of the POI,

- $long_1$and$g_2$are the longitudes of the POI,
- Δlat = (lat_2- lat_1) is the difference in latitude,
- $\Delta long$ = ($long_1$ - $long_1$) is the difference in longitude

3.3.1.4 SPPUR Similarity Formula

To obtain the final **SPPUR similarity score** between each pair of users, we combine the three individual similarity components Simpath, Simstart, and Simend using the combination parameters α, β, and γ, as defined in Formula (3.8) below:

$$\boldsymbol{SPPUR} = \alpha\left(Sim_{path}(u_a, u_b)\right) + \beta\left(Sim_{start}(u_a, u_b)\right) + \gamma\left(Sim_{end}(u_a, u_b)\right) \tag{3.8}$$

Where:

- α, β, and γ are weighting coefficients used to adjust the contribution of each similarity component,
- α, β, γ ∈ [0,1] and α+β+γ=1

3.3.1.5 SPPUR Prediction Formula

Once the SPPUR similarity values between users have been computed using Formula (3.8), the prediction of potential POI visits for a target user is obtained using the following formula [80].

$$Predict(user_u, POI_i) = \frac{\sum_{v \in U} Sim(u,v) \times f_{v,i}}{\sum Sim(u,v)} \tag{3.9}$$

With:

- U is the set of all users.
- $Sim(u,v)$ is the final similarity (SPPUR) between users u and v.
- $f_{v,i}$ is the visit frequency of $user_v$ on POI_i

3.3.2 Example of SPPUR similarity calculation

In the following, we calculate SPPUR similarities from Table 1 below, which contains a dataset of four users and five POIs.

Table 3.1 An example of user-POI frequency matrix

Users	POI_1	POI_2	POI_3	POI_4	POI_5
$User_1$	1	2	0	4	1
$User_2$	3	0	1	0	2
$User_3$	0	0	4	1	6
$User_4$	3	0	1	0	0

1- Calculation of the TF Value :

To compute the TF value for each pair (Usera, POIi), we use the formula (3.1). For instance, the TF value for the pair (User1, POI1) is calculated as follows:

$$TF(User_1, POI_1) = \frac{1}{1+2+4+1} = 0.125$$

Similarly, we calculate the rest of the TF pair values as shown in Table 3.2:

Table 3.2 TF score matrix

TF	POI_1	POI_2	POI_3	POI_4	POI_5
$User_1$	0.125	0.25	0	0.5	0.125
$User_2$	0.5	0	0.166	0	0.333
$User_3$	0	0	0.363	0.09	0.545
$User_4$	0.75	0	0.25	0	0

2- Calculating the IDF value :

To calculate the IDF value of each POI, we use formula (3.2). For example, the IDF value of POI1 is calculated as follows:

$$IDF(POI_1) = log_2(^4/_3) = 0.415$$

In the same way, we calculate the other IDF values of the POIs as shown in Table 3.3

Table 3.3 IDF score matrix

	POI_1	POI_2	POI_3	POI_4	POI_5
IDF	0.415	2	0.415	1	0.415

3- Calculating the TF-IDF value :

To calculate the TFI-DF value of each pair (Usera,POIi), we use formula (3.3). For example, the TFI-DF value of this pair (User1, POI1) is calculated as follows:

$$TFIDF(User_1, POI_1) = TF(User_1, POI_1) \times IDF(POI_1) = 0.051$$

In the same way, we calculate the other values of the TF pairs as shown in Table 3.4:

Table 3.4 TF-IDF score matrix

TF-IDF	POI_1	POI_2	POI_3	POI_4	POI_5
$User_1$	0.051	0.5	0	0.5	0.051
$User_2$	0.207	0	0.069	0	0.138
$User_3$	0	0	0.150	0.090	0.226
$User_4$	0.311	0	0.103	0	0

4- Calculating $\mathbf{SIM_{path}}$ similarity :

Finally, to calculate the simpath similarity between each pair (Usera, Userb), we use formula (3.4). For example, the simpath value between User1 and User2 is calculated as follows:

$$\boldsymbol{Sim_{path}(User_1, User_2)} = \frac{(0.051 \times 0.207) + (0.5 \times 0) + (0 \times 0.069) + (0.5 \times 0) + (0.051 \times 0.138)}{\sqrt{(0.051)^2 + (0.5)^2 + (0.5)^2 + (0.051)^2} \times \sqrt{(0.207)^2 + (0.069)^2 + (0.138)^2}}$$

$$\boldsymbol{Sim_{path}(User_1, User_2)} = 0.0975$$

In the same way, we calculate the other similarity values between the different user pairs, as shown in Table 3.5:

Table 3.5 $\mathrm{Sim_{path}}$ matrix

$\mathbf{Sim_{path}}$	$User_1$	$User_2$	$User_3$	$User_4$
$User_1$	1	0.0975	0.2805	0.0692
$User_2$	0.0975	1	0.5624	0.8452
$User_3$	0.2805	0.5624	1	0.1664
$User_4$	0.0692	1	0.1664	1

5- Calculating $\mathrm{Sim_{start}}$ and $\mathrm{Sim_{end}}$ similarities:

Considering the scenario described in Figure 3.1 and Figure 3.2, user 1 and user 2 have POI_1/POI_3 as starting points and POI_4/POI_1 as arrival points.

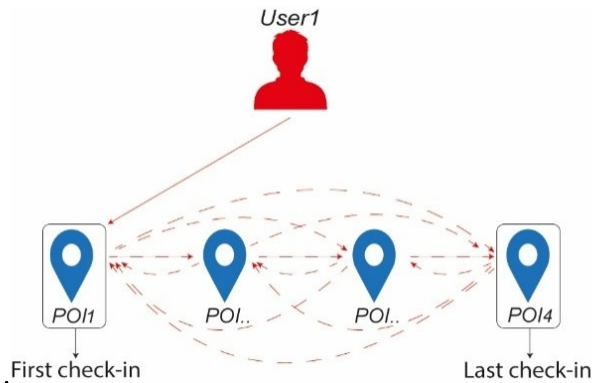

Figure 3.1 path taken by user 1

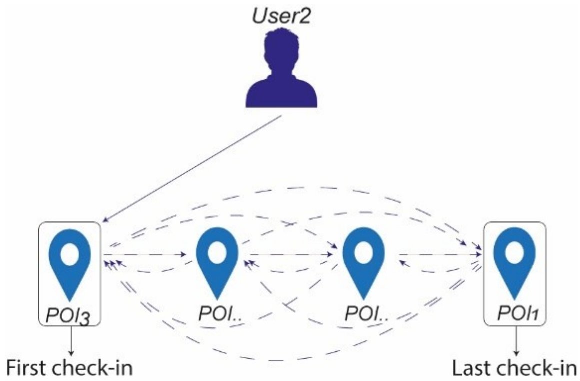

Figure 3.2 path taken by user 2

Based on the POI distance matrix (see Table 3.6), which describes the different distances between POI pairs calculated using formula (3.7), we can deduce the Sim_{start} and Sim_{end} similarities between each user pair using formulas (3.5) and (3.6) respectively.

Table 3.6 POIs distance

	POI_1	POI_2	POI_3	POI_4	POI_5
POI_1	0	1	6	3	2
POI_2	1	0	2	5	5
POI_3	6	2	0	1	3
POI_4	3	5	1	0	2
POI_5	2	5	3	2	0

For example, to calculate the Sim_{start} similarity between user1 and user2, we use formula 3.5 as shown below:

$$Sim_{start}(user_1, user_2) = \frac{1}{1 + dist_{start}(user_1, user_2)} = \frac{1}{1 + dist(POI_1, POI_3)} = \frac{1}{1+6}$$

$Sim_{start}(user_1, user_2)$ =0.1428

Furthermore, to calculate the Sim_{end} similarity between $user_1$ and $user_2$, we use formula 3.6 as shown below:

$$Sim_{end}(user_1, user_2) = \frac{1}{1 + dist_{end}(user_1, user_2)} = \frac{1}{1 + dist(POI_4, POI_1)} = \frac{1}{1+3}$$

$Sim_{send}(user_1, user_2) = 0.25$

Similarly, we calculate the rest of the Sim_{end} similarity and Sim_{start} similarity values between the different user pairs, as shown in Tables 3.7 and 3.8:

Table 3.7 Sim_{start} matrix

$\mathbf{Sim_{start}}$	*User1*	*User2*	*User3*	*User4*
User1	1	0.1428	0.25	1
User2	0.1428	1	0.5	0.1428
User3	0.25	0.5	1	0.25
User4	1	0.1428	0.25	1

Table 3.8 Simend matrix

$\mathbf{Sim_{end}}$	*User1*	*User2*	*User3*	*User4*
User1	1	0.25	0.5	0.333
User2	0.25	1	0.1428	0.333
User3	0.5	0.1428	1	0.25
User4	0.333	0.333	0.25	1

6- Calculating SPPUR similarity :

Finally, to calculate SPPUR similarity, we use formula (3.8). For example, the SPPUR similarity between user 1 and user 2 can be deduced using the following formula :

$$\boldsymbol{SPPUR}(user_1, user_2) = \alpha\left(Sim_{path}(user_1, user_2)\right) + \beta\left(Sim_{start}(user_1, user_2)\right) + \gamma\left(Sim_{end}(user_1, user_2)\right)$$

For example, for the values $\alpha = 0.5$, $\beta = 0.25$ et $\gamma = 0.25$, we get the result below:

$$\boldsymbol{SPPUR}(user_1, user_2) = 0.5 \times (0.0975) + 0.25 \times (0.1428) + 0.25 \times (0.25)$$

$$\boldsymbol{SPPUR}(user_1, user_2) = 0.1469$$

Similarly, we calculate the other SPPUR similarity values between the different user pairs as shown in Table 3.9:

Table 3.9 SPPUR matrix

$\mathbf{Sim_{end}}$	*User1*	*User2*	*User3*	*User4*
User1	1	0.1469	0.3277	0.3679
User2	0.1469	1	0.4419	0.5416
User3	0.3277	0.4419	1	0.2082
User4	0.3679	0.5416	0.2082	1

3.3.3 SPPUR model Algorithm

In this section, we present a pseudocode for Algorithm 1 below, which will be used to implement the model named SPPUR. This algorithm is designed to compute the SPPUR similarities between users and subsequently predict the POIs to be visited.

Algorithm 1 : SPPUR

Input : R : Users-POIs Frequency-Matrix ; U : Vector of users ; P : Vector of POIs ;
TF : Vector of TF value of each pair*(user i, POI p)*; *IDF* : Vector of IDF value;
FirstP : Vector of the first POI visited for each user ;
LastP : Vector of the last POI visited for each user ;
N : Number of similar users ; *NSusers* : N most similar users for the target user;
K : Total number of POIs to be recommended ;
UId : Id of the target user ; a, β, γ : Ajustment value ;
Output : *SPPURM* : users final similarity matrix ;
PPM : POIs prediction matrix ;
ListRecPOI : List of POIs to be recommended ;

```
//similarity between users using formula (3.8) ;
for each user U(i) do
    for each user U(j) ≠ U(i) do
        N=0 ; Di=0, Dj=0 ;
        for each POI p do
            If (R(i,p) ≠ 0 or R(j,p) ≠ 0) then
                Di = Di + (TF(i,p) × IDF(p))^2 ;
                Dj = Dj + (TF(j,p) × IDF(p))^2 ;
                N=N+(Di × Dj) ;
        USpathM(i,j) = N / (√Di × √Dj);
        USstartM(i,j) = 1 / (1+dist(FirstP(i),FirstP(j)));
        USendM(i,j) = 1 / (1+dist(LastP(i),LastP(j)));
        SPPURM(i,j) = a(USpathM(i,j)) + β(USstartM(i,j)) + γ(USendM(i,j)) ;
//Select N most similar users for the target user;
NSusers = Descending sorting(SPPURM(i,j), UId, N) ;
//Prediction computation using formula (3.9) ;
for each user i do
    for each POIp do
        N=0 ; D=0 ;
        for each NSusers(j) do
            N=N+( SPPURM(i,j) × R(j,p)) ;
            D = D + SPPURM(i,j);
        PPM(i,j) = N / D;
//Select K POIs for the target user;
ListRecPOI = Descending sorting(PPM, UId, K) ;
return ListRecPOI ;
```

3.3.4 Time Complexity Analysis of the SPPUR Algorithm

The time complexity of the SPPUR algorithm can be analyzed by examining its main computational components :

A. User Similarity Computation:

The algorithm computes pairwise similarities between all users based on their interactions with POIs. This involves iterating over each pair of users and, for each pair, over all POIs. Assuming m users and n POIs, this step has a complexity of :

$$O(m^2 * n)$$

B. Most Similar Users Selection:

For each target user, the algorithm sorts the similarity scores to select the top $\boldsymbol{N}$ similar users. Sorting m scores for each user leads to a complexity of :

$$O(m^2 * \log m)$$

C. Prediction Computation:

For each user and each POI, the algorithm aggregates scores from the $\boldsymbol{N}$ most similar users. This results in a complexity of :

$$O(m * n * N)$$

If N is considered constant or significantly smaller than m, this step becomes:

$$O(m * n)$$

D. Recommendation Ranking:

The algorithm sorts the predicted POIs to select the top $\boldsymbol{K}$ recommendations. For each user, sorting over n POIs has a complexity of :

$$O(n * \log n)$$

Thus, for all users :

$$O(m * n * \log n)$$

Overall Complexity:

Combining all the components, the total time complexity of the ***SPPUR*** algorithm is:

$$O(m^2 * n + m^2 * \log m + m * n * N + m * n * \log n)$$

Assuming $\boldsymbol{N}$ is a small constant, the dominant term is :

$$O(m^2 * n)$$

This reflects the quadratic cost in the number of users due to the pairwise similarity computation, which is typical in user-based collaborative filtering approaches.

3.4 The SPPUR model

The SPPUR model is based on the SPPUR similarity, which is inspired by the TF-IDF method. This model leverages the frequency and geographical proximity between POIs and users to calculate similarities, thereby enabling the prediction of POIs that may be of interest to new tourists. The main steps of the proposed model are outlined below:

Step 1: Preprocessing phase: First, from the existing dataset, we construct the User-POI frequency matrix (see Figure 3.3). Then, we normalize the check-in frequency of each user into the range [0, 5]. The process of normalization is described as follow:

$$POI_{Nfreq} = \begin{cases} 5, if\ POI_{freq} = Max_{freq} \\ \dfrac{5 \times POI_{freq}}{Max_{freq}}, otherwise \end{cases} \tag{3.10}$$

Where POI_{Nfreq}indicates the normalized frequency value, POI_{freq} indicates the real check-in number of a user, Max_{freq} indicate the largest frequency of a user.

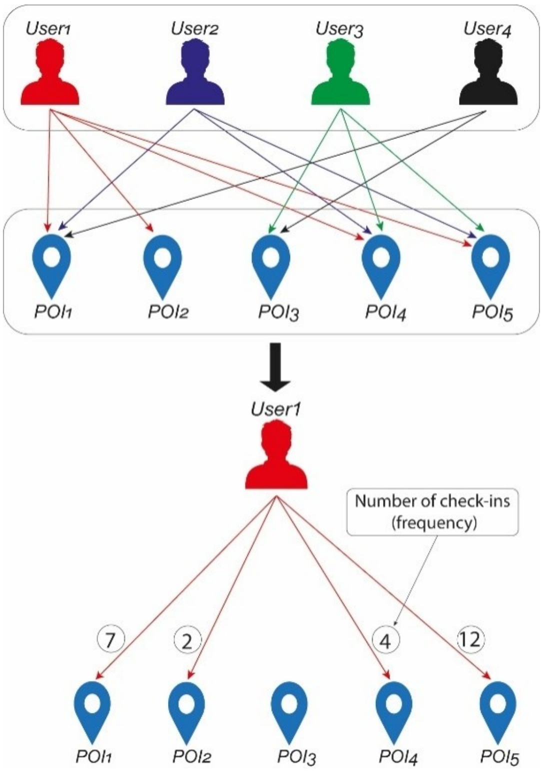

Figure 3.3: Transition from the user check-in to the POI frequentation

Step 2: for the target user ($user_a$), we calculate the TFI-DF score of all pairs ($User_a, POI_i$), then, we compute the Sim_{path} similarity between the target user and each other user ($User_b$) by using formula (3.4).

Step 3: After computing the $\mathbf{Sim_{path}}$, $\mathbf{Sim_{start}}$ and $\mathbf{Sim_{end}}$, we use the formula (3.8) to compute the final ***SPPUR*** similarity between the target user and the other users.

Step 4: Next, a list of N most similar users to the target user is selected.

Step 5: Then a prediction is generated by using the formula (3.9).

Step 6: Finally, we recommend a Top K ranked POIs to the target user.

Figure 3.4 illustrates the different steps of our Framework based on the SPPUR model.

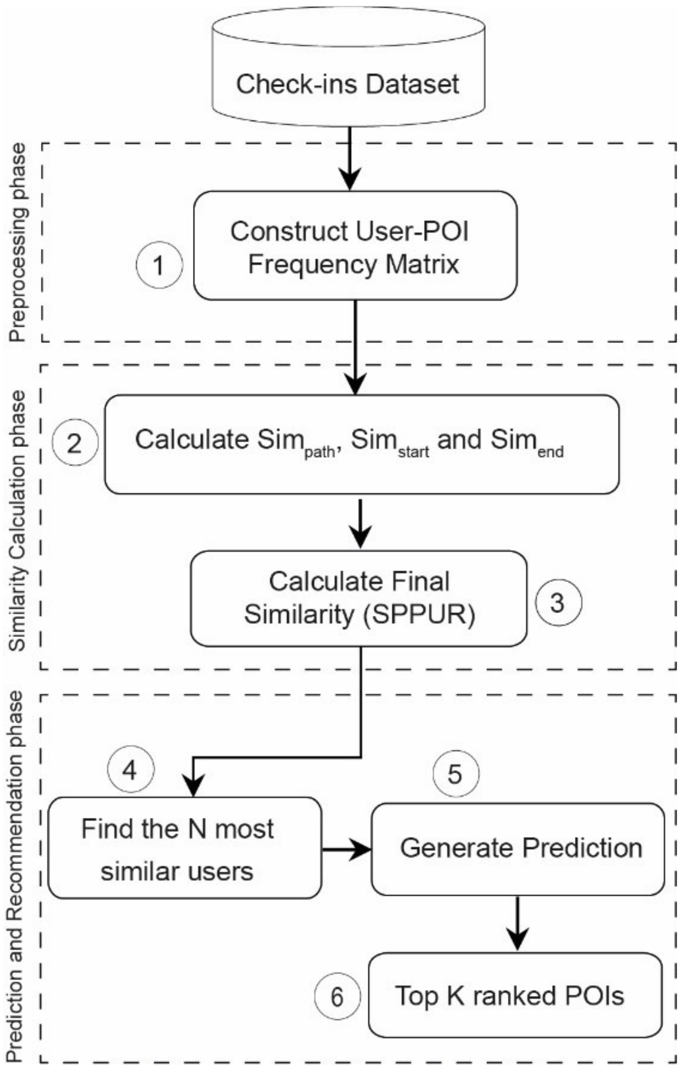

Figure 3.4 SPPUR model Framework

3.5 Experiments

In this section, we described the Foursquare dataset utilized in our experiments. Subsequently, we explained the evaluation metrics (PRECISION, RECALL, MAP, and NDCG) selected to assess the performance of the SPPUR model. Finally, we outlined our experimental procedure and the associated hyper parameters.

3.5.1 Data Collection

For the implementation of the SPPUR model, we used the Foursquare dataset, which contains check-ins from tourists associated with Points of Interest (POIs). In our study, we focused on data from the cities of New York and Tokyo, covering the period from April 12, 2012, to February 16, 2013. Each record in this dataset includes the check-in time, the user ID, the POI ID, the POI's longitude, the POI's latitude, and the POI category ID [110]. Table 3.10 presents a detailed description of the columns in the Foursquare dataset used in our experiments.

Table 3.10 Dataset used in the experiments

Dataset	**New York Dataset**	**Tokyo Dataset**
Users	1083	2293
POIs	38333	61858
Check-ins	227428	573703

3.5.2 Evaluation metrics

To compare the SPPUR model with algorithms based on other types of similarity existing in the literature, such as Pearson correlation [111] [112], Spearman correlation [16] [113], Euclidean distance [113], cosine similarity [16], adjusted cosine similarity [16], Mean Squared Error (MSE or MSD) [111], and Jaccard similarity measure [113], we employed four evaluation metrics: Precision, Recall, MAP (Mean Average Precision), and NDCG (Normalized Discounted Cumulative Gain).

3.5.3 Hyper Parameters settings

The hyper parameters used of all experiments were defined as shown in Table 3.11:

Table 3.11 hyper parameters

Symbol	Description
$\boldsymbol{K}$	Number of recommended POIs (Top ranked POI)
$\boldsymbol{N}$	Number of similar users
%Training set	Dataset used to produce recommendation
%Test set	Dataset used on evaluation phase
a, β, γ	SPUUR similarity adjustment values

To conduct experiments, 80% of the data in the datasets are used as training data, while 20% are used as testing data to evaluate the accuracy of the methods. We recommend for each user 5, 10, 15 and 20 top ranked POIs ($\boldsymbol{K}$=5,10,15,20) by considering 50 similar users ($\boldsymbol{N}$=50). The optimal values of a, β and γ are: (a = 0.75, β = 0.075 et γ = 0.175). In addition, experiments were conducted using PHP on a Windows 11 64-bit operating system with an Intel(R) Core (TM) i5-8th@1.30 GHz processor and 20 GB RAM.

3.5.4 Experimental procedure

We evaluated the performance of our SPPUR model by comparing it to other recommendation models using traditional similarity measures such as Pearson's correlation, Spearman's correlation, Euclidean distance, cosine, adjusted cosine, mean square error (MSD or MSE) and Jaccard's similarity measure. To do this, we divided the dataset into a training data and the test data in terms of check-ins. We use 80% of the check-ins records generated by each user for the training data and the rest for the test data. After, we use user-based collaborative filtering method to recommend Top@K (K=5, 10, 15, 20) POIs to each user. Finally, to evaluate the performance of the algorithm used, we used PRECISION, RECALL, MAP and NDCG. The evaluation steps are described below:

1) Divide the Dataset into training data and test.
2) Build the User-POI frequency matrix.
3) For each user:
   - Calculate the similarity between the selected user and the other users.
   - Select the ***N*** most similar users to the selected user.
   - Generateprediction.
   - Select ***K*** Top-ranked POIs to recommend to the selected user.
   - Compute the PRECISION@k, RECALL@k, MAP@k and NDCG@k.
4) Compute the global PRECISION@k, RECALL@k, MAP@k and NDCG@k.

3.6 Results and Discussion

To test the SPPUR model, we utilized two Foursquare datasets: the first dataset pertains to the city of New York, and the second dataset pertains to the city of Tokyo. Initially, we compared the different variants of the SPPUR model to identify the one that yields the best values for the parameters PRECISION, RECALL, MAP, and NDCG. Subsequently, we compared the best-performing variant of the SPPUR model with various state-of-the-art models that employ similarity principles for POI recommendation. Finally, we analyzed and discussed the results obtained during the evaluation process of the SPPUR model.

3.6.1 Comparison Between the Three Variants of SPPUR Model

Initially, we assess the performance of each similarity variant proposed within the SPPUR model. The precision@5, recall@5, MAP@5, and NDCG@5 scores for these similarity types are presented in order to determine the optimal similarity measure. Figure 3.5 and Figure 3.6 compare the values of PRECISION@5, RECALL@5, MAP@5, and NDCG@5 for (1) the Sim_{path} similarity, which is based solely on the POI paths, (2) the SimStartEnd similarity, which relies only on the start and arrival points of the visits, and finally, (3) the SPPUR similarity, which combines the Sim_{path}, Sim_{Start}, and Sim_{End} similarities as defined in subsection 3.1. The results of this comparison demonstrate that the SPPUR similarity outperforms both the Simpath and SimStartEnd similarities in terms of PRECISION, RECALL, MAP, and NDCG. Consequently, we utilize the SPPUR similarity for the SPPUR model, as it also allows for the adjustment of the combination of the Simpath, SimStart, and SimEnd similarities through the hyperparameters α, β, and γ.

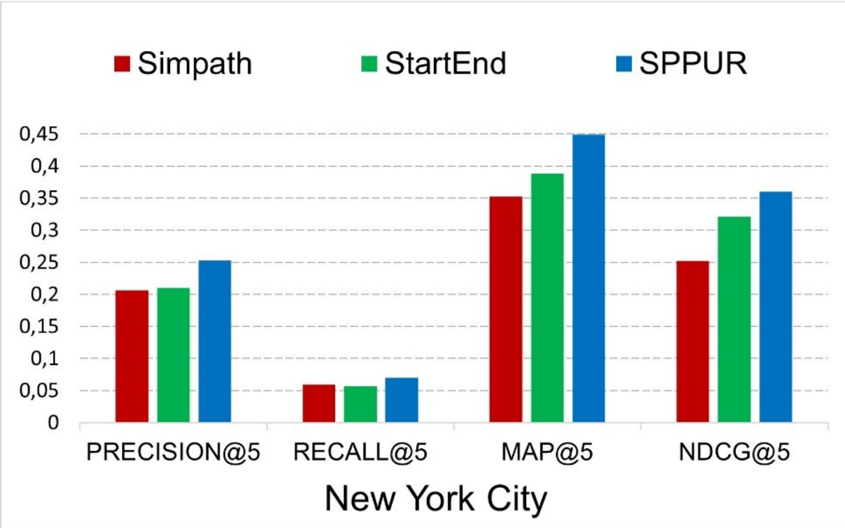

Figure 3.5 Comparison of different variants of SPPUR by using New York dataset.

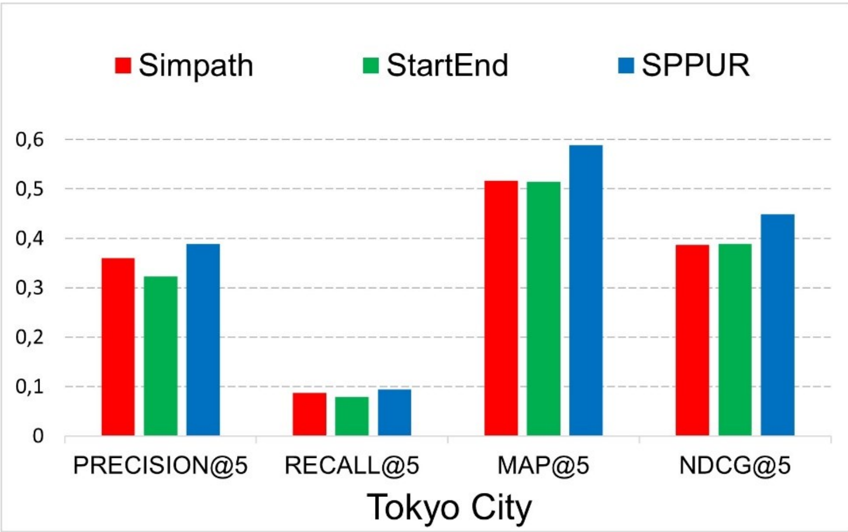

Figure 3.6 Comparison of different variants of SPPUR by using Tokyo city dataset.

3.6.2 Comparison Between SPPUR Model and Other Similarity

In the following section, we compare our model based on the SPPUR similarity measure with other models from the literature that utilize traditional similarity measures, such as Pearson correlation, Spearman correlation, Euclidean distance, cosine similarity, adjusted cosine similarity, mean squared error (MSE or MSD), and the Jaccard similarity index. This comparison focuses on the parameters of PRECISION, RECALL, MAP, and NDCG, using the datasets from the city of New York and the city of Tokyo.

Figure 3.7 and Figure 3.9 below illustrate that the PRECISION of all models using similarity measures decreases as the number of recommended POIs (K) increases. However, Figure 3.8 and Figure 3.10 demonstrate that RECALL improves as the number of recommended POIs decreases.

On the other hand, Figure 3.7, Figure 3.8, Figure 3.9, and Figure 3.10 indicate that the PRECISION and RECALL metrics of the SPPUR model outperform those of other similarity-based models in both the New York and Tokyo datasets..

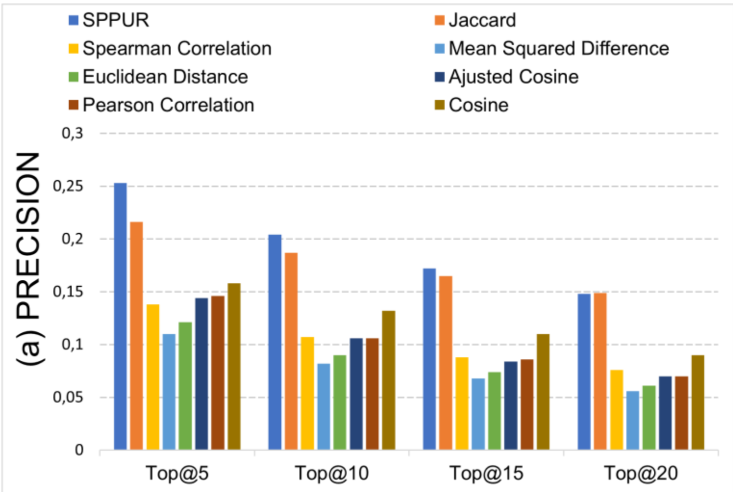

Figure 3.7 PRECISION performance on the New York City data set

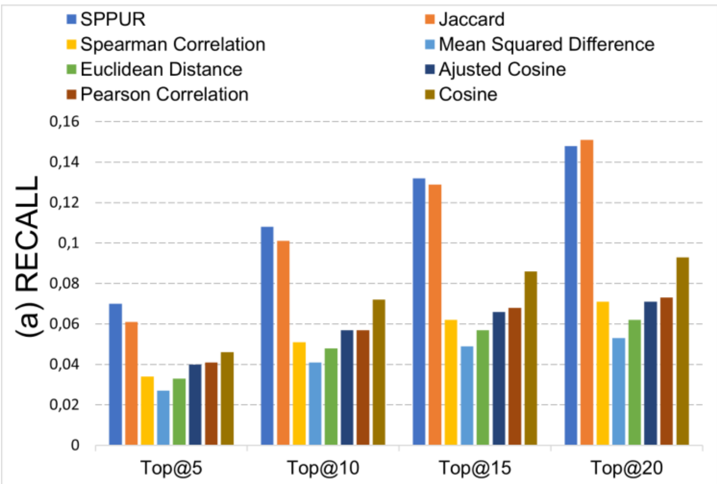

Figure 3.8 RECALL performance on the New York City data set

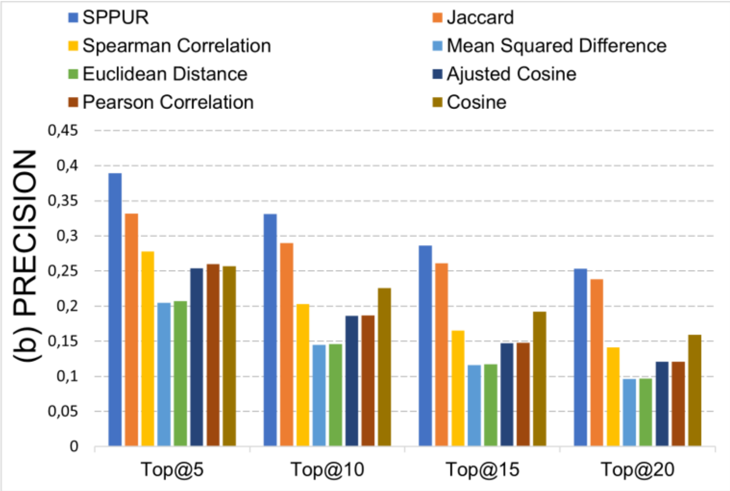

Figure 3.9 PRECISION performance on the Tokyo City data set

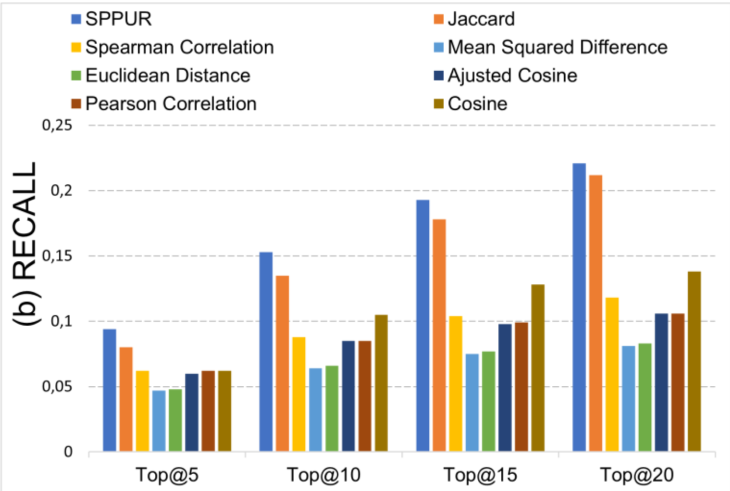

Figure 3.10 RECALL performance and the Tokyo City data set

Figures 3.11 and 3.13 presented below demonstrate that the Mean Average Precision (MAP) of all models utilizing similarity measures decreases as the number of recommended Points of Interest (POIs) (K) increases. However, Figures 3.12 and 3.14 show that the Normalized Discounted Cumulative Gain (NDCG) increases as the number of recommended POIs decreases.

On the other hand, Figures 3.11, 3.12, 3.13, and 3.14 indicate that the MAP and NDCG metrics of the SPPUR model outperform those of other similarity-based models when applied to the New York and Tokyo datasets.

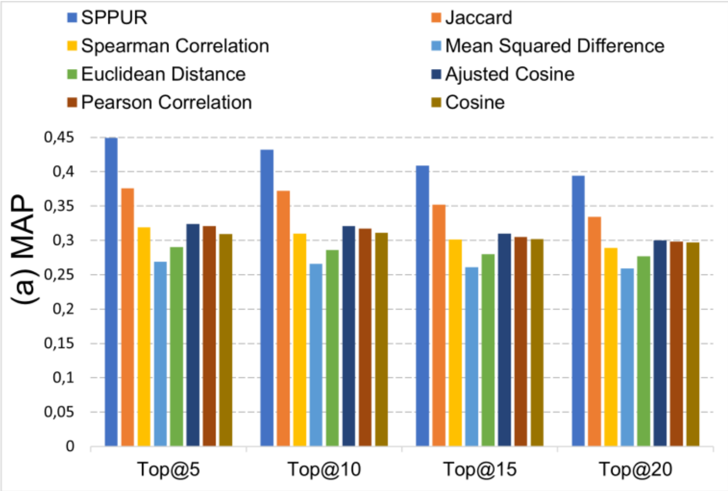

Figure 3.11 MAP performance and the New York City data set

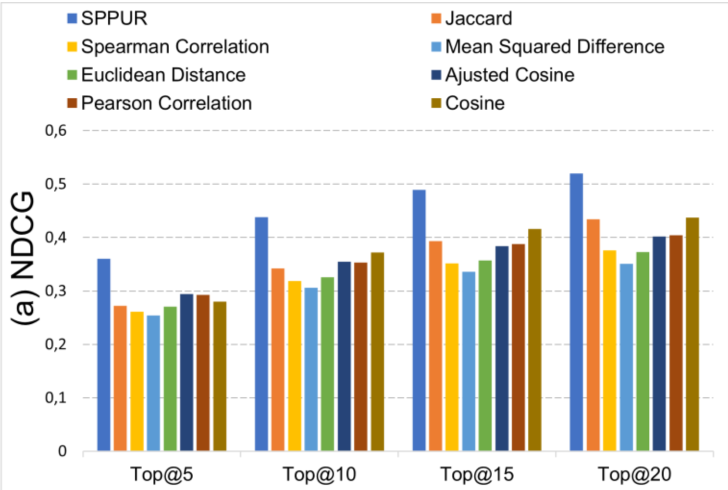

Figure 3.12 NDCG performance and the New York City data set

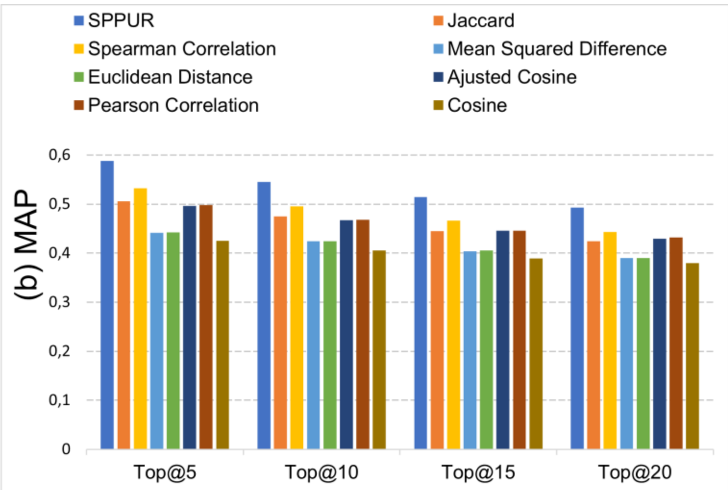

Figure 3.13 MAP performance and the Tokyo City data set

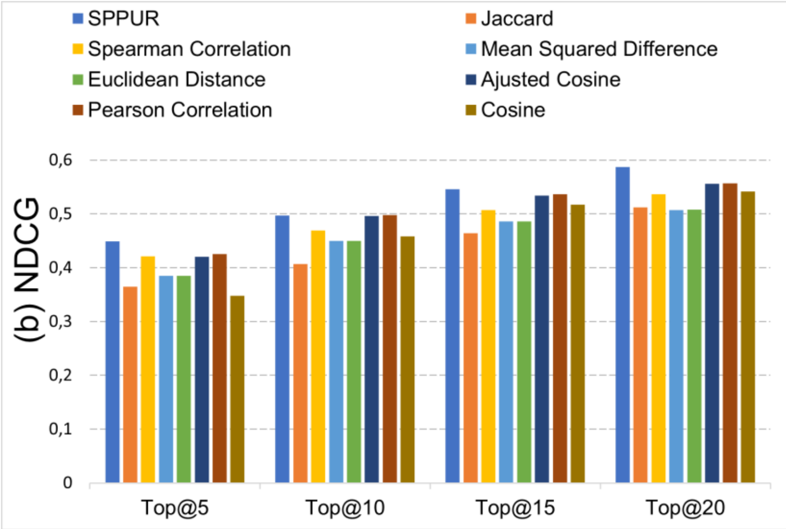

Figure 3.14 NDCG performance and the Tokyo City data set

Finally, the SPPUR similarity measure employed in our model appears to outperform traditional measures in terms of precision, recall, MAP, and NDCG. These experiments demonstrate the effectiveness of this novel similarity measure, as evidenced by its performance on two distinct datasets representing two different cities (New York and Tokyo).

3.6.3 Results summary

The results presented in subsection 3.6.1 demonstrate that the SPPUR similarity outperforms other variants (Simpath, SimStart, and SimEnd) in terms of precision, recall, MAP, and NDCG. Consequently, this similarity measure is selected for the SPPUR model. In subsection 3.6.2, this model is compared with other models from the literature that employ traditional similarity measures such as Pearson correlation, Spearman correlation, Euclidean distance, cosine similarity, adjusted cosine similarity, mean squared error (MSE or MSD), and Jaccard similarity. This comparison is conducted using precision, recall, MAP, and NDCG metrics on the New York and Tokyo datasets. Ultimately, the results confirm the effectiveness of the SPPUR similarity measure, as it consistently outperformed traditional similarity measures.

3.6.4 The effect of the neighborhood size in SPPUR model

To examine the effect of the neighborhood size (denoted as N in the algorithm of subsection 3.3.3) on the performance of the SPPUR model, we recommended the Top@KPOIs for each user by varying the neighborhood size used in the user-user similarity calculation. The performance of the SPPUR model is evaluated based on different neighborhood sizes (***N***) in terms of precision, recall, MAP, and NDCG. The results of these evaluations are presented in Table 3.12 below.

Table 3.12 Comparison results on different Number of similar users

	New York Dataset				**Tokyo Dataset**			
N	PRECISION	RECALL	MAP	NDCG	PRECISION	RECALL	MAP	NDCG
5	0,229	0,06	**0,487**	**0,485**	**0,419**	0,098	**0,651**	**0,566**
10	0,249	0,067	0,486	0,457	0,418	**0,099**	0,636	0,53
20	0,255	0,069	0,466	0,406	0,411	0,098	0,62	0,497
30	**0,257**	**0,07**	0,461	0,384	0,401	0,097	0,61	0,477
40	0,254	**0,07**	0,451	0,369	0,396	0,096	0,599	0,463
50	0,253	**0,07**	0,449	0,36	0,389	0,094	0,588	0,449

The results obtained indicate that an increase in the number of neighbors leads to a decrease in the MAP and NDCG values. This suggests that the selection of the number of neighbors plays a significant role in the calculation of the SPPUR similarity.

3.6.5 Discussions

In recent years, the rapid development of location-based social networks has significantly facilitated the tourism activities of individuals. Similarity measures play a critical role and can have a substantial impact on improving the accuracy of recommendations for places to visit. In this paper, we propose a novel similarity measure based on the TF-IDF technique for user-based point of interest (POI) recommendation systems. This measure leverages the effectiveness of TF-IDF in the analysis and utilization of check-ins generated by tourists during their visits to POIs. Experimental results demonstrate that the proposed similarity measure enhances the accuracy of POI recommendation systems utilizing user-based collaborative filtering, as it outperforms traditional methods. Finally, from a future perspective, we aim to further improve our approach by incorporating additional contextual factors such as the semantic characteristics of POIs, geographical regions, weather conditions, and seasonal variations.

3.7 Conclusion

This chapter introduced SPPUR, a novel similarity measure for Point of Interest (POI) recommendation systems, which enhances existing approaches by incorporating both user check-ins and travel trajectories. Unlike traditional methods that rely solely on individual visits, our model leverages movement sequences and geographical proximity to more accurately predict tourists' preferences.

Experiments conducted on the Foursquare dataset validated the effectiveness of SPPUR, demonstrating a significant improvement in performance compared to traditional similarity measures (such as Cosine and Pearson). Specifically, our approach addresses two major challenges in recommendation systems: data sparsity and the cold-start problem, while maintaining high accuracy as evaluated through metrics like Precision, Recall, MAP, and NDCG.

The results open several avenues for future work:

- **Extension to other contexts**: Adapting SPPUR to other types of urban mobility (e.g., transportation, events).
- **Integration of temporal data**: Incorporating visit timings to refine recommendations.
- **Optimization of similarity calculation**: Reducing algorithmic complexity for real-time applications.

Chaptre 4 The IPUMC model

4.1 Introduction

This chapter introduces a new model, named **IPUMC** (Integrating Proximity of Users in Modified Cosine similarity), which combines the implicit similarity between users based on their check-ins with their geographic proximity [114]. The goal is to enhance the relevance of recommendations by considering both users' behavioral preferences and their spatial context. After defining the problem, we will elaborate on the various components of the model, including the methods for calculating similarities, before presenting the IPUMC model itself. An experiment using a real-world dataset will validate the proposed approach and analyze the obtained results.

4.2 Problem Definition

The rise of Location-Based Social Networks (LBSNs) has generated an increasing demand for Point of Interest (POI) recommendation systems capable of delivering relevant and personalized suggestions to users. These systems generally rely on Collaborative Filtering (CF) techniques, which identify similar users or items by analyzing past interactions. However, traditional CF methods, such as those based on Pearson correlation, cosine similarity, or Euclidean distance, exhibit significant limitations when applied to LBSNs.

The main issue lies in the inability of conventional similarity measures to capture the complexity of interactions specific to LBSNs. In particular, these approaches overlook two crucial dimensions :

- **Geographical Dimension** : Users tend to frequent POIs that are geographically close to their current or past locations, a factor that is insufficiently accounted for in traditional models.

- **Social Dimension** : Users' preferences are often influenced by their social network, a factor that is either underutilized or treated independently in existing algorithms.

Moreover, recent approaches attempting to incorporate these dimensions (geographical and social) often do so in an indirect manner, treating them as auxiliary components or post-processing steps. This separation undermines the coherence and relevance of recommendations, especially in dense urban environments where the simultaneous consideration of these factors is essential.

Therefore, the primary challenge is to design an innovative similarity measure capable of directly integrating both geographical proximity and social relationships into the collaborative filtering process. Such a measure would enhance the contextual relevance of POI recommendations, taking into account not only users' personal preferences but also their geographic location and social connections. This approach seeks to outperform traditional methods and address user expectations in complex real-world scenarios. This chapter aims to achieve three main objectives :

1. **Development of a New Similarity Measure Integrating Geographical Proximity :**

This similarity measure combines traditional metrics, such as Pearson correlation and cosine similarity, with the direct incorporation of geographical proximity between users. This extension aims to better capture the dynamics specific to location-based social networks (LBSNs), where user interactions are significantly influenced by their geographical location. By accounting for both social behaviors (such as ratings or check-ins) and the spatial distance between individuals, this hybrid measure provides a more nuanced evaluation of similarity, thereby enhancing the relevance of recommendations.

2. **Evaluation of the Effectiveness of This Measure in a CF Framework :**

The proposed similarity measure will be implemented within a collaborative filtering-based recommendation system to experimentally validate its effectiveness. This will be done using two real-world datasets from Foursquare, covering densely populated cities such as New York and Tokyo.

3. **Comparison of the New Method's Performance with Classical Approaches :**

To assess the performance of the new similarity measure in comparison to traditional methods (Pearson correlation, Spearman correlation, cosine similarity), we employ robust evaluation metrics such as Precision, Recall, F1-Score, Mean Average Precision (MAP), and Normalized Discounted Cumulative Gain (NDCG).

4.3 Problem Formulation

Our approach focuses on calculating the similarity between users in a Points of Interest (POI) recommendation system, leveraging their check-in histories and combining two types of similarity:

- Similarity Based on Check-in History :

This method examines user profiles through their historical visits to POIs. It employs a specially calibrated version of cosine similarity, adapted to account for visit frequencies, in order to measure the degree of overlap between the various users' tourist paths.

- Similarity Based on the Order of Check-ins :

We hypothesize that the similarity between two users can be evaluated based on the geographical proximity of their first and last check-ins. To formalize this relationship, we introduce a geographic distance measure (GeoDist) that precisely quantifies this spatial closeness.

The **IPUMC similarity** integrates these two measures to produce a hybrid similarity, thereby optimizing the recommendation process. The performance of this approach is evaluated using real-world data (Foursquare, densely populated cities), and is compared against traditional methods to validate its effectiveness. The following section explains how behavioral similarity (visit history) and temporal geographic proximity (order of check-ins) can be combined to make POI recommendations.

4.3.1 User-User Similarity Calculation Based on Check-ins

This section investigates user profiles through their historical tourist check-ins. Our approach is based on the hypothesis that the similarity between users can be assessed by analyzing the common Points of Interest (POIs) they visit during their travels. To quantify this similarity, we use an adapted version of cosine similarity, where (1) each POI is weighted by its relative visit frequency, and (2) the score is normalized by dividing the visit frequency of each POI by the total number of visits of the user. This method enables the establishment of an accurate similarity measure between pairs of users, according to the following mathematical formulation :

$$MCos(U_a, U_b) = \frac{\sum_{i \in I} \left(W_{(U_a)}(P_i) \times W_{(U_b)}(P_i) \right)}{\sqrt{\sum_{i \in I} \left(W_{(U_a)}(P_i) \right)^2} \times \sqrt{\sum_{i \in I} \left(W_{(U_b)}(P_i) \right)^2}} \tag{4.1}$$

Where:

- I represent the set of $POIs$ visited by $User_a$ and $User_b$.
- $W_{(U_a)}(P_i) = Freq_{(U_a)}(P_i) / \, Max_{freq}(U_a)$

4.3.2 Calculation of User-to-User Similarity Based on Check-in Order

We hypothesize that two users, U_a and U_β, exhibit behavioral similarity when their starting and ending points (the first and last visited Points of Interest, or POIs) are geographically close. This spatial proximity of their trajectories is quantified using our GeoDist metric, defined as follows:

$$GeoDist(U_a, U_b) = \frac{\alpha}{1 + dis(FP_a, FP_b)} + \frac{\beta}{1 + dis(LP_a, LP_b)} \quad (4.2)$$

Where:

- $\alpha, \beta \in [0,1]$ and $\alpha + \beta = 1$
- $dis(FP_a, FP_b)$is the distance between the initial POIs visited by users U_a *and* U_b.
- $dis(LP_a, LP_b)$represents the distance between the last POIs visited by users U_a and U_b.

Note that to calculate the distance between two points of interest (POI), we used the Haversine formula donner ref, which is commonly employed in navigation and geographical information systems.

4.3.3 Calculating IPUMC similarity

To calculate the IPUMC similarity between each pair of users, we combine the MCos and GeoDis values using the parameters γ and δ, as shown in the equation below.

$$IPUMC = \gamma\big(Mcos(u_a, u_b)\big) + \delta\big(GeoDist(u_a, u_b)\big) \quad (4.3)$$

Where:

- $\gamma, \delta \in [0,1]$ and $\gamma + \delta = 1$

4.3.4 The IPUMC Model

This section provides a detailed description of the IPUMC model's architecture. It organizes the primary steps of the recommendation system, which includes the integration of IPUMC similarity to assess user affinities. The global architecture of the system is visualized in Figure 4.1.

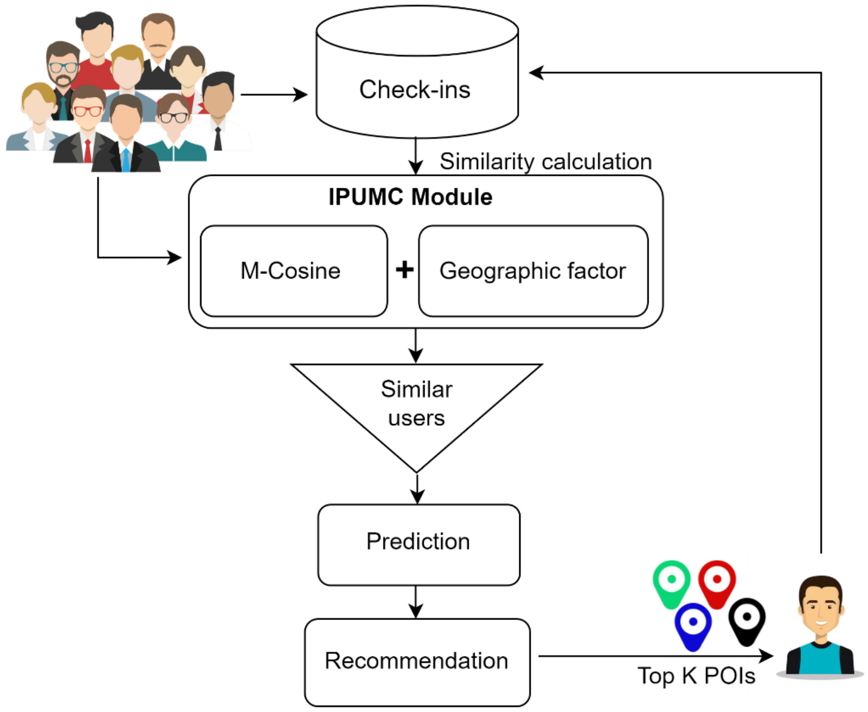

Figure 4.1 The working mechanism of RS using IPUMC model

Algorithm 1 above shows the pseudocode of the IPUMC model :

Algorithm 1: IPUMC Model

Input: R: users-POIs check-ins matrix, $UserID$: target user;

Output:$ListTopPOI$;

Begin

Step 1: Data processing (Transform the user check-ins into a User- POI visit matrix);

Step 2: Compute the similarity between $UserID$ and the other users;

Step 3: Select $\boldsymbol{N}$ the most similar users to $UserID$;

Step 4: Generate all POI predictions for the user $UserID$;

Step 5: Select top $\boldsymbol{K}$ POIs;

Step 6: Return $ListTopPOI$ for $UserID$;

End.

4.4 Experimentation

In this section, we introduce the Foursquare dataset utilized in our experiments, followed by a detailed explanation of the evaluation metrics (PRECISION, RECALL, F1-score, MAP, and NDCG) chosen for performance analysis of IPUMC. The experimental methodology implemented in this study is also thoroughly described.

4.4.1 Data collection

The implementation of the IPUMC model is based on the Foursquare dataset, comprising geolocated tourist check-ins on POIs in New York and Tokyo. The detailed characteristics of this dataset are summarized in Table 4.1.

Table 4.1 Dataset used in the experiments.

	New York Dataset	Tokyo Dataset
Users	1083	2293
POIs	38333	61858
Check-ins	227428	573703

4.4.2 Parameters

After conducting extensive experimental tests, the parameters of our system have been established as follows:

- $\alpha = 0.6$, $\beta = 0.4$, $\gamma = 0.85$ and $\delta = 0.15$ for New York Dataset.
- $\alpha = 0.6$, $\beta = 0.4$, $\gamma = 0.8$ and $\delta = 0.2$ for Tokyo Dataset.
- N = 30 and K = [5, 10, 15, 20] for both Datasets.

4.4.3 Experimental Protocol

The performance evaluation of the IPUMC model was conducted by comparing it against various traditional similarity metrics: Pearson correlation, Spearman correlation, Euclidean distance, cosine similarity, adjusted cosine similarity, and Jaccard similarity. The dataset was partitioned into a 70/30 split between the training and testing sets, based on the historical user check-ins.

A user-centered collaborative filtering approach was implemented to generate Top@K POI recommendations. The comparative analysis relied on the following metrics: PRECISION, RECALL, F1-score, MAP, and NDCG.

The experimental protocol specified the following:

1. A 70/30 data split (training/testing)
2. Recommendation of the top 5, 10, 15, and 20 POIs
3. Selection of the 30 most similar neighbors for each user.

4.4.4 Results and Discussion

Our model, based on the IPUMC similarity measure, is compared to reference approaches utilizing traditional measures such as Pearson correlation, Spearman correlation, Euclidean distance, cosine similarity, adjusted cosine similarity, and Jaccard similarity. This evaluation is based on standard performance metrics: PRECISION, RECALL, F1-score, MAP (Mean Average Precision), and NDCG (Normalized Discounted Cumulative Gain). The tests were conducted on the New York and Tokyo datasets to ensure a robust and comparative validation of the model's performance.

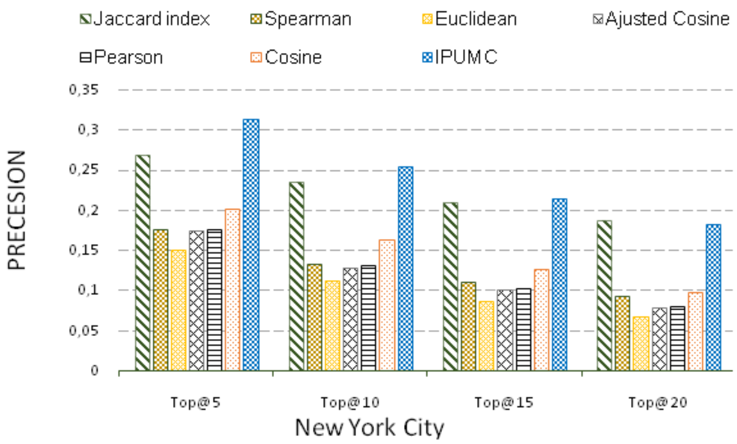

Figure 4.2 PRECESION performance on New York City

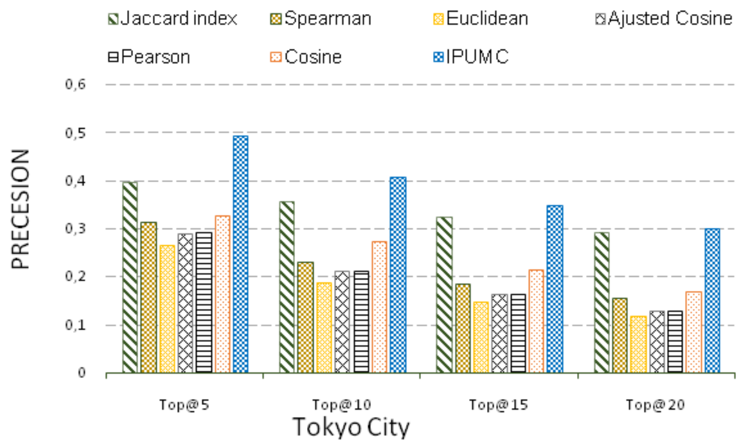

Figure 4.3 PRECESION performance on Tokyo City

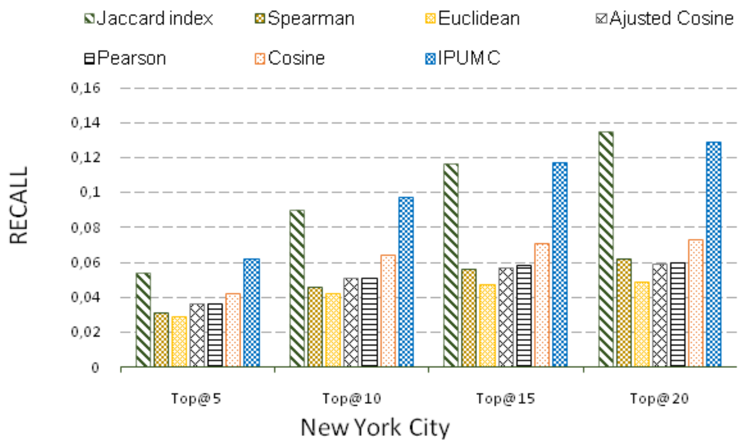

Figure 4.4 RECALL performance on New York City

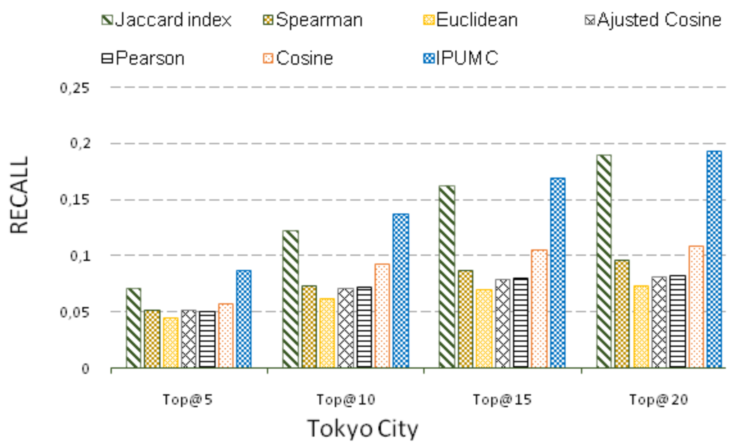

Figure 4.5 RECALL performance on Tokyo City

Figure 4.2, Figure 4.3 illustrate that the PRECISION values decrease as the number of recommended POIs (K) increases for both datasets (NYC and Tokyo City). In contrast, Figure 4.4 and Figure 4.5 demonstrate that the RECALL values increase with the number of recommended POIs. Additionally, Figure 4.2, Figure 4.3, Figure 4.4, and Figure 4.5 indicate that the PRECISION and RECALL achieved with the IPUMC similarity outperform those of the other metrics across both datasets.

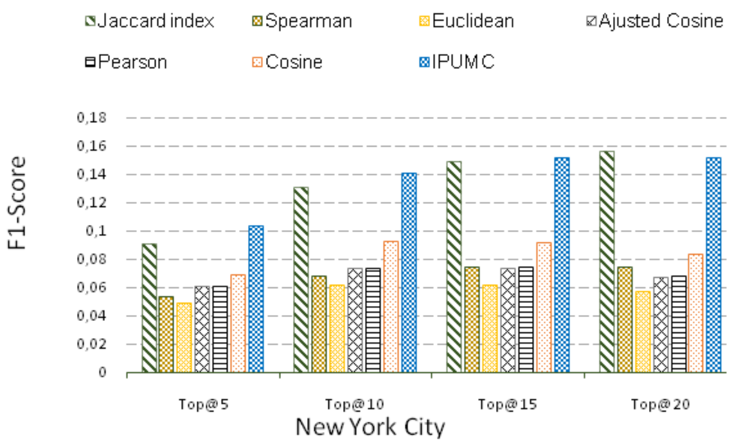

Figure 4.6 F1-score performance on New York City

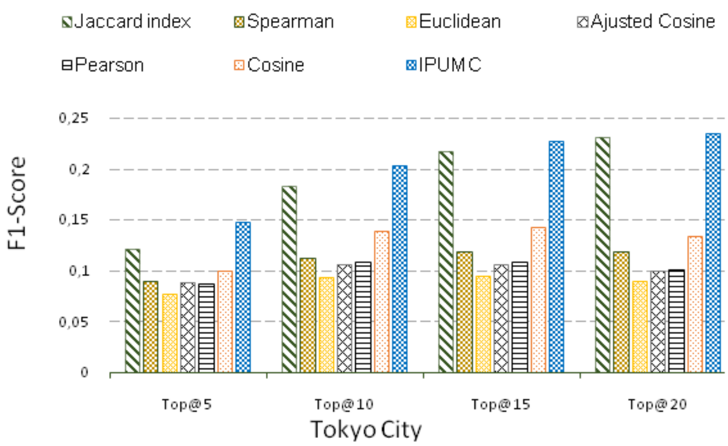

Figure 4.7 F1-score performance on Tokyo City

Figure 4.6 and Figure 4.7 show that F1 scores increase for all similarity measures as K increases. It also shows that the F1 score of the IPUMC model is higher than the other measures in both datasets.

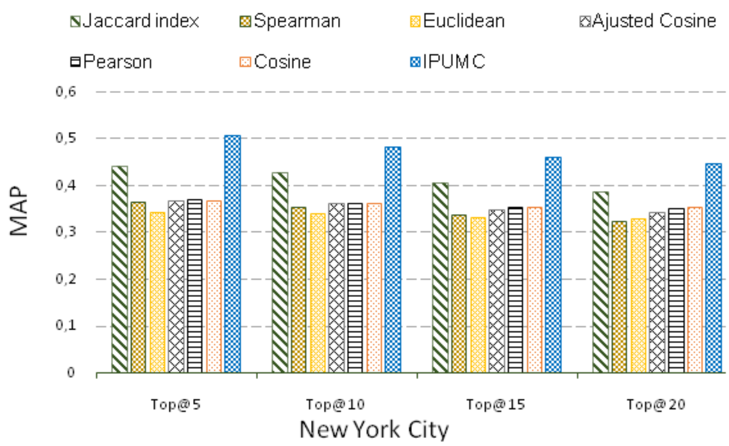

Figure 4.8 MAP performance on New York City

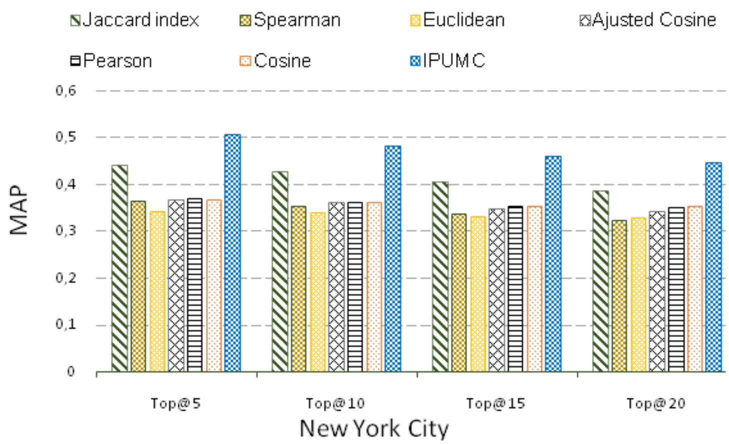

Figure 4.9 MAP performance on Tokyo City

Figure 4.8 and Figure 4.9 illustrate that the MAP values decrease as K increases, while Figure 4.10 and Figure 4.11 show that the NDCG values rise when the number of recommended POIs decreases.

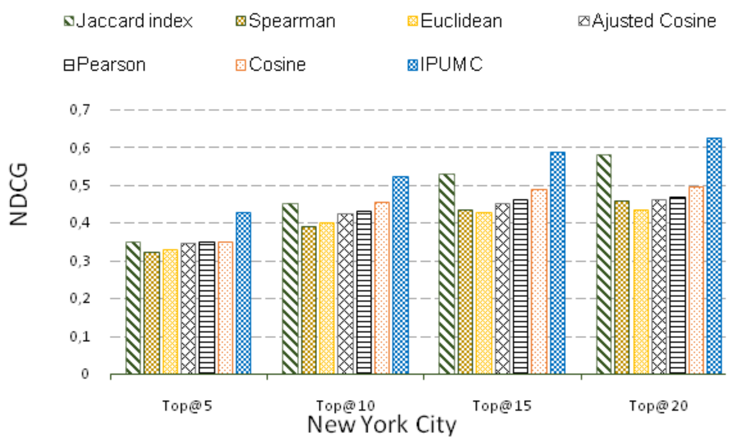

Figure 4.10 NDCG performance on New York City

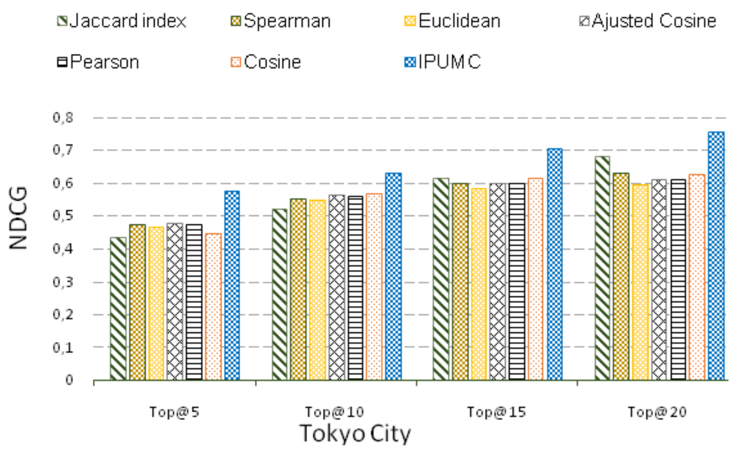

Figure 4.11 NDCG performance on Tokyo City

In conclusion, the IPUMC similarity measure, evaluated on the New York and Tokyo datasets, demonstrated superior performance compared to traditional metrics (Pearson correlation, Spearman correlation, Euclidean distance, cosine similarity, adjusted cosine similarity, and Jaccard similarity) in terms of precision, recall, F1 score, MAP, and NDCG (see Figure 4.2 to Figure 4.11).

4.5 Conclusion

This chapter presented the **IPUMC** model (Implicit Proximity and User Matching via Check-ins), a hybrid approach that integrates both implicit similarity between users and their geographical proximity, aiming to enhance the quality of Point of Interest (POI) recommendations in location-based social networks. By combining information derived from check-ins with spatial distances between users, the model succeeds in better capturing behavioral and contextual affinities.

After clearly defining the problem, we introduced various methods for calculating similarity, including the order of check-ins, to construct a richer user-to-user proximity indicator. The experiment conducted on a real-world dataset allowed for an evaluation of the model's performance, highlighting a significant improvement compared to traditional approaches that overlook the geographical dimension.

The results confirm the relevance of integrating spatial proximity into collaborative filtering mechanisms, paving the way for future improvements, including the integration of other contextual factors such as time or dynamic preferences. The next chapter will explore the potential for incorporating more geographical context into recommendation systems.

Chaptre 5 The IUPJS model

5.1 Introduction

In the continuation of efforts to improve the quality of point-of-interest (POI) recommendations in location-based social networks (LBSNs), this chapter introduces a novel model named IUPJS (Implicit User Proximity and Jaccard-based Similarity). This model is distinctive in that it explicitly incorporates the spatial dimension specifically, the users' location into the similarity calculation between profiles [115]. Unlike traditional collaborative filtering approaches, which rely solely on interaction histories (such as ratings or check-ins), the IUPJS method combines implicit behavioral similarity with geographical proximity to identify users with similar profiles.

The chapter begins with a precise definition of the problem under study, followed by a mathematical formulation of the proposed method. It then details the process for calculating IUPJS similarity and the method for predicting recommended POIs. The structure of the model is formally presented, after which the experimental phase is introduced, relying on real-world data sourced from Foursquare. The results obtained are then analyzed using standard evaluation metrics such as precision, recall, F1-score, MAP, and NDCG. This analysis demonstrates the effectiveness of the IUPJS model compared to traditional approaches, both in terms of recommendation relevance and the incorporation of spatial context.

5.2 Problem Definition

With the rapid rise of location-based social networks (LBSNs), point-of-interest (POI) recommendation systems have become essential for helping users discover new, relevant locations. These systems primarily rely on users' personal preferences and geographic proximity to generate tailored recommendations. Among the most widely explored techniques in this field is collaborative filtering (CF), particularly methods based on similarities between users or between items, as illustrated in the figure below.

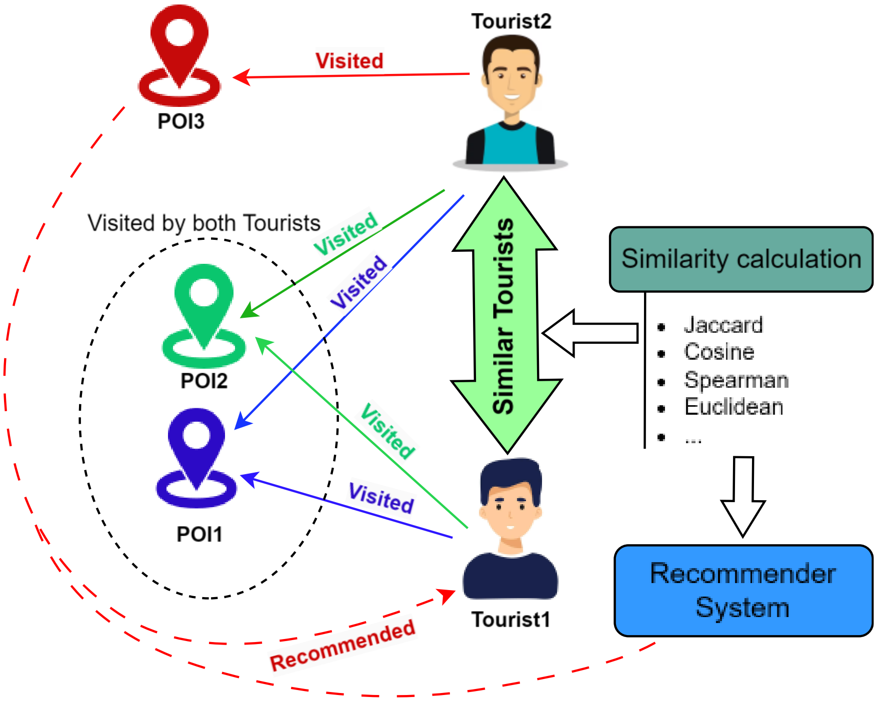

Figure 5.1 User-based Collaborative filtering technique

Collaborative filtering based on users is grounded in the assumption that users exhibiting similar behaviors or preferences can benefit from cross-recommendations. These systems rely on conventional similarity measures such as Pearson correlation, cosine similarity, or Euclidean distance, which are computed from user ratings or check-ins. However, the effectiveness of these approaches is often constrained by their inability to account for essential contextual factors in Location-Based Social Networks (LBSNs), such as geographic proximity and social influence between users.

Recent studies have emphasized the importance of incorporating these contextual dimensions to enhance recommendation relevance. Indeed, users tend to visit locations that are close to their current or past position, and their choices are frequently influenced by the behaviors of their social network. Despite these observations, existing models still often treat geographic and social proximity as secondary information, failing to integrate them directly into the core similarity calculation.

5.2.1 Problem Statement

Traditional collaborative filtering models exhibit limitations in geolocated environments as they do not explicitly account for the spatial and social dimensions inherent to users. Consequently, it becomes essential to reconsider similarity measures by incorporating both geographic proximity and behavioral similarity between users, in order to generate more personalized and context-aware recommendations.

5.2.2 Objectives

In this context, this work proposes a novel similarity measure that combines both behavioral and geographical dimensions to enhance the quality of Points of Interest (POI) recommendations. This measure is evaluated using real-world data from Foursquare and compared to conventional collaborative filtering approaches that employ standard similarity measures (Pearson, cosine, Jaccard, etc.), based on metrics such as precision, recall, F1-score, MAP, and NDCG.

5.3 Problem Formulation

This section outlines the methodology for calculating the IUPJS similarity measure, which is based on users' check-in history, and its application in predicting Points of Interest (POIs). In this context, the formulas used to compute this similarity are explained in detail, along with those employed to generate predictions from the similarity scores obtained.

5.3.1 Calculation of the IUPJS Similarity

The IUPJS similarity measure is constructed by combining two complementary types of similarity. The first, inspired by the Jaccard index, quantifies and normalizes the degree of overlap between the sets of POIs visited by users. The second type is based exclusively on initial check-ins, reflecting users' initial choices at the start of their journeys. These two measures are then integrated to produce an overall similarity, referred to as IUPJS, which forms the foundation for the prediction process within the POI recommendation system.

This approach relies on analyzing user profiles constructed from their check-in histories during their tourist visits. The central hypothesis is that the similarity between two users can be assessed based on the overlap of locations they have visited. Thus, similarity is calculated using the Jaccard index, applied to the sets of POIs. The formulas for measuring this similarity between two users are presented below :

$$IUPJS(u_a, u_b) = \alpha \times \frac{|I_a \cap I_b|}{|I_a \cup I_b|} + (1 - \alpha) \times GeoF(u_a, u_b) \qquad (5.1)$$

Where,

$$GeoF(u_a, u_b) = \beta \cdot \frac{1}{1 + First(u_a, u_b)} + \delta \cdot \frac{1}{1 + Last(u_a, u_b)} \qquad (5.2)$$

With:

- I_a and I_b : sets of pois visited by users u and u_b, respectively.
- $GeoF(u_a, \mathrm{u_b})$: geographic influence component between u_a and u_b.
- $First(\mathrm{u_a}, \mathrm{u_b})$: euclidean distance between the first pois visited by u_a and u_b.
- $Last(\mathrm{u_a}, \mathrm{u_b})$: euclidean distance between the last pois visited by u_a and u_b.
- $\alpha \in [0,1]$, $\beta, \delta \in [0,1]$, and $\beta + \delta = 1$: balances the contributions of shared preferences and geographic influence

5.3.2 Prediction calculation

To calculate POI predictions, we used the prediction formula below [80]. This formula is based on the IUPJS similarity.

$$Predict(user_u, POI_i) = \frac{\sum_{v \in U} Sim(u, v) \times f_{v,i}}{\sum Sim(u, v)} \qquad (5.3)$$

Where :

- U represents the set of users,

- Sim(u,v) denotes the final similarity (IUPJS) between users u and v,

- fv,i is the number of visits by user v to POIi.

5.4 The IUPJS Model

The IUPJS model is a recommendation process based on a series of structured steps : First, a User-POI matrix is constructed from the check-in dataset, representing the interactions between users and the places they visit. Once this matrix is established, it is appropriately partitioned to distinguish between training and testing sets. For a given target user, the IUPJS similarity scores are then computed for all pairs by comparing the target user's profile with those of other users. Based on this, a list of the N most similar users is identified. Predictions for POIs likely to interest the target user are then generated. Finally, a list of the K most relevant points of interest is presented, constituting the final recommendation.

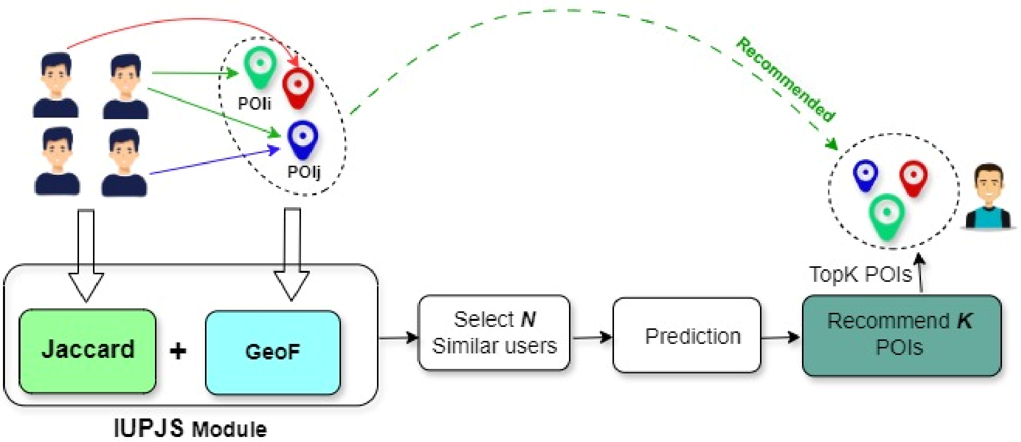

Figure 5.2 The main steps of the IUPJS model

The main steps of the IUPJS model can be summarized as follows :

Step 1 **:** Constructing a User-POI matrix based on the check-in dataset.

Step 2 **:** Proper partitioning of the User-POI matrix after its construction.

Step 3 **:** For the target user u_a, compute the IUPJS scores for all pairs (u_a, u_b).

Step 4 **:** Identifying a list of the ***N*** most similar users.

Step 5 **:** Generating predictions using Equation (5.3).

Step 6 **:** Recommend to the target user a list of the ***K*** most relevant POIs.

5.5 Experiments and Results

This section presents the Foursquare dataset used for the experiment, followed by a detailed description of the evaluation metrics namely PRECISION, RECALL, F1-score, MAP, and NDCG which are employed to assess the performance of the IUPJS model.

5.5.1 Data Collection

To evaluate the proposed similarity measure, we utilized a Foursquare check-in dataset collected in the city of Tokyo [110]. This dataset exhibits a high level of sparsity, with users having visited only a small fraction of the available Points of Interest (POIs). Table 5.1 summarizes the key characteristics of the dataset used.

Table 5.1 Foursquare Dataset (Tokyo city)

Dataset	**Tokyo Dataset**
Users	2293
POIs	61858
Check-ins	573703
Sparsity (%)	99.85

5.5.2 Comparison with Basic Approaches

The performance of our method was evaluated by comparing it with several reference techniques, which represent classical solutions for similarity and recommendation in collaborative filtering. Therefore, we contrasted our approach with different widely used reference methods in recommendation systems based on collaborative filtering and similarity calculation. These methods can be summarized as follows:

- **POP:** A baseline model recommending the most popular Points of Interest (POIs) to users.
- **CF-JACC:** User-centered collaborative filtering using the Jaccard index, which measures the overlap of visited POIs between two users, focusing on behavioral similarity.
- **CF-SPEAR:** User-centered collaborative filtering using Spearman's similarity, a rank-based measure that accounts for the ordinal nature of user preferences.
- **CF-EUCL:** User-centered collaborative filtering using Euclidean distance, a measure that quantifies the spatial separation between users' check-in or preference vectors.
- **CF-COS:** User-centered collaborative filtering using cosine similarity, which measures the cosine of the angle between users' feature vectors, regardless of their magnitude.

5.5.3 Main Hyperparameters

Following a series of rigorous experiments, we have determined the optimal hyperparameters for our system. The selected values are as follows: $\alpha = 0.75$, $\beta = 0.3$, $\delta = 0.7$, $N = 40$, and $K = [5, 10, 15]$. This selection is based on a comparative analysis aimed at maximizing the model's performance.

5.5.4 Evaluation Protocol

As part of the comparative evaluation of the IUPJS model, we adopted a stringent experimental protocol. This protocol involves assessing its performance against several classic similarity algorithms, including Spearman's correlation, Euclidean distance, cosine similarity, and the Jaccard index. For this analysis, five standard metrics were chosen : Precision, Recall, F1-score, as well as the MAP (Mean Average Precision) and NDCG (Normalized Discounted Cumulative Gain) measures. This approach enables a multidimensional evaluation of the relevance and robustness of the proposed model.

5.5.5 Experimentation of Our Evaluation Protocol

To assess the performance of our IUPJS method, we conducted a comparative analysis with several classical similarity measures, including Pearson correlation, Spearman correlation, Euclidean distance, cosine similarity, and the Jaccard index. The dataset was partitioned according to a standard split of 90% for training and 10% for testing. The methodology employed relies on a user-centered collaborative filtering approach, with Top@K recommendations being evaluated using five main metrics: PRECISION, RECALL, F1-score, MAP (Mean Average Precision), and NDCG (Normalized Discounted Cumulative Gain).

This experimental process can be summarized in four main steps:

1) **Data Preparation**: segmentation of check-ins into training set (90%) and tesset (10%).
2) **Modeling** : Construction of the User-POI matrix.
3) **Calculation of similarity between users based on the selected measure**. Selection of the 40 nearest neighbors (***N***) for each target user. Generation of predictions and extraction of the top K rated POIs (***K*** {5,10,15}). Evaluation metric calculation for each recommendation.
4) **Synthesis**: Aggregation of metric results across all users to obtain an overall performance measure.

Finally, this experiment systematically applied a configuration of 40 similar users (N=40) and three recommendation lengths (K=5, 10, 15), allowing for a granular analysis of the impact of the recommendation list size.

5.6 Results and Discussion

In our experiments with the IUPJS model, we utilized two Foursquare datasets: the New York dataset and the Tokyo dataset. The IUPJS model was compared to various baseline models that apply similarity-based methods for POI recommendation. Finally, we analyzed and discussed the results obtained during the evaluation process of the IUPJS model.

We compare our model, based on the IUPJS similarity measure, with baseline models from the literature that use traditional similarity measures such as Pearson correlation, Euclidean distance, cosine similarity, and Jaccard similarity. This comparison focuses on metrics including **PRECISION**, **RECALL**, **F1-score**, **MAP**, and **NDCG**, using the Foursquare dataset from Tokyo.

This analysis highlights the ability of **IUPJS** to integrate contextual dimensions (both geographical and behavioral), while outperforming conventional approaches, thus validating its effectiveness for accurate and personalized POI recommendations.

Figure 5.3 and Figure 5.6 show that **PRECISION** and **MAP** decrease as the number of recommended POIs (K) increases. In contrast, Figure 5.4, Figure 5.5, and Figure 5.7 reveal that **RECALL**, **F1-score**, and **NDCG** improve as K decreases. Furthermore, Figure 5.3 to Figure 5.7 clearly demonstrate that the **PRECISION**, **RECALL**, **F1-score**, **MAP**, and **NDCG** parameters of the IUPJS model consistently outperform those of other baseline models.

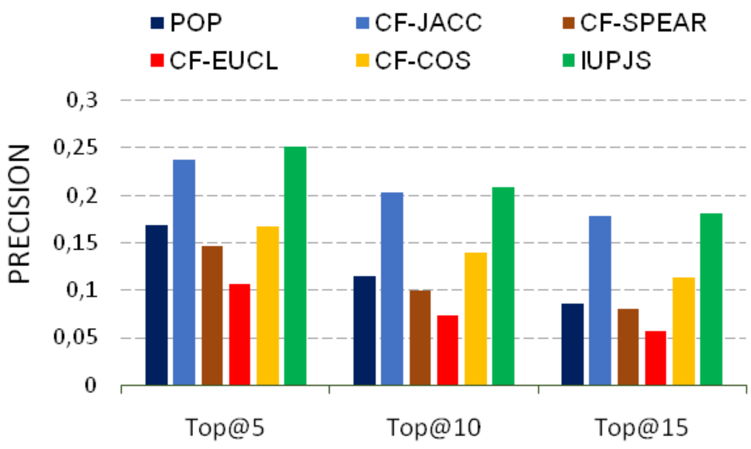

Figure 5.3 PRECISION performances

On the one hand, Figure 5.3, Figure 5.4, Figure 5.5, Figure 5.6, and Figure 5.7 demonstrate that the precision, recall, F1-score, MAP, and NDCG metrics of the IUPJS model outperform those of the baseline models.

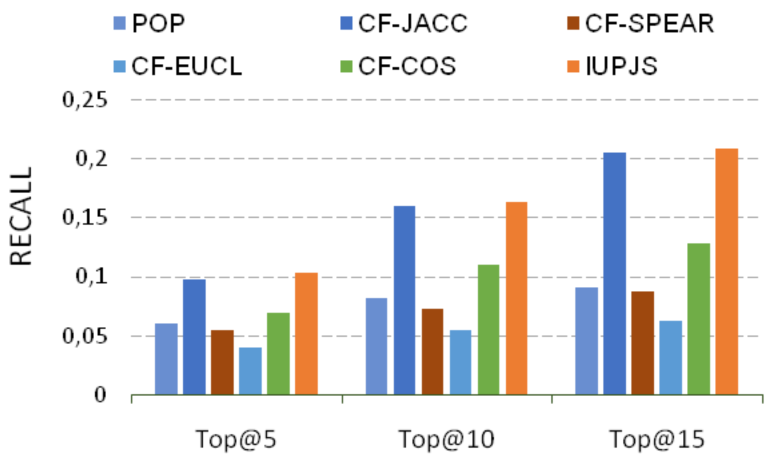

Figure 5.4 RECALL performances

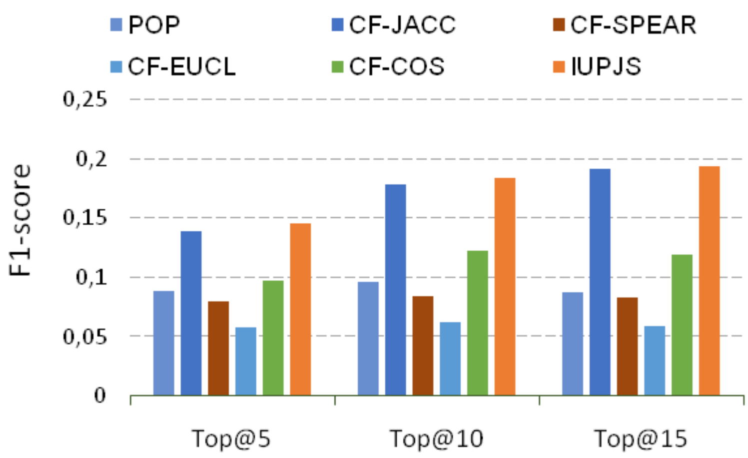

Figure 5.5 F1-score performances

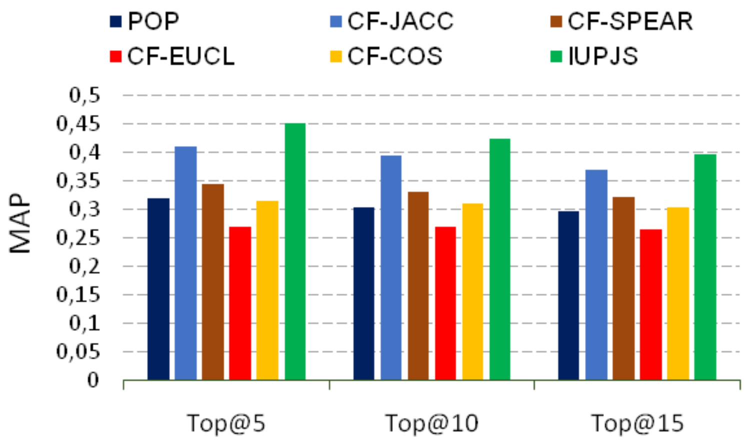

Figure 5.6 MAP performances

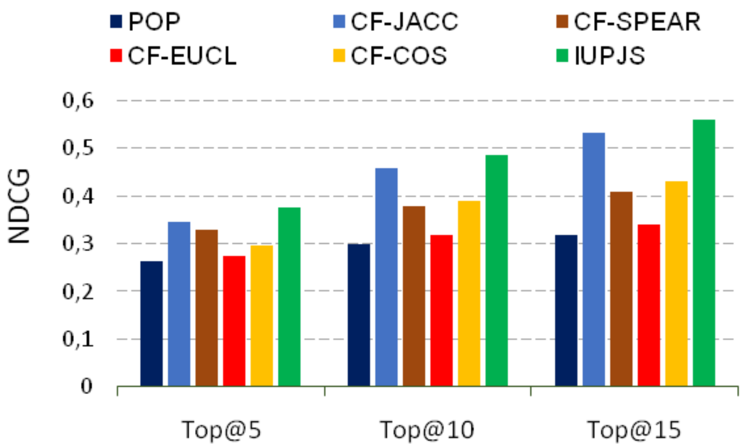

Figure 5.7 NDCG performances

In conclusion, the IUPJS similarity measure, calculated using the Tokyo dataset, has shown superior performance compared to the baseline methods in terms of precision, recall, F1-score, MAP, and NDCG.

The experimental results demonstrate that the IUPJS similarity measure significantly outperforms traditional similarity-based reference methods in terms of performance. This notable improvement is attributed to its ability to effectively integrate both geographical influence and historical user check-in behavior into the similarity calculation. By combining these two critical factors, the IUPJS-based approach generates recommendations that are both accurate and contextually relevant. This enhanced capability to simultaneously consider geographical location and user behavior history emphasizes its potential to improve user satisfaction through recommendations better aligned with individual preferences. Such an approach holds particular promise in areas like smart tourism, where a tourist's location and visit history are crucial for delivering personalized and meaningful Points of Interest (POI) recommendations.

The results also highlight the superiority of our method, which synergistically combines geographical proximity and user behavior, compared to traditional approaches that rely solely on conventional similarity measures. This comparison underscores the importance of considering contextual factors in order to improve the quality of POI recommendation systems..

5.7 Conclusion

This chapter introduced and detailed the IUPJS model, an innovative approach to recommending Points of Interest (POIs) that combines implicit user similarity with geographical proximity. By integrating these two dimensions into the similarity calculation, IUPJS addresses the limitations of traditional collaborative filtering methods, which are often insensitive to the spatial context of users. After defining and formalizing the problem, we described the methodology for calculating IUPJS similarity, as well as the prediction formulas used to recommend POIs. An experimental phase, conducted with real data from Foursquare, allowed for the evaluation of the model's performance based on several standard metrics. The results highlighted the superiority of the IUPJS model over traditional approaches, both in terms of accuracy and the geographical relevance of recommendations. These findings pave the way for new perspectives in the improvement of contextual recommendation systems, further exploiting implicit spatial and behavioral dimensions.

General Conclusion

This thesis is part of an effort to rethink Point of Interest (POI) recommendation mechanisms by leveraging innovative similarity measures capable of simultaneously capturing users' preferences and their spatial context. Three major contributions have been proposed: the SPPUR, IPUMC, and IUPJS models, each addressing specific limitations of traditional approaches.

1) **The SPPUR model** introduced a similarity measure inspired by the TF-IDF technique, combining users' movement sequences with their geographical proximity. Experiments conducted on Foursquare datasets (New York and Tokyo) demonstrated that SPPUR outperforms traditional methods, particularly in mitigating data sparsity and cold-start issues. This approach opens promising perspectives for extending to other types of urban mobility and incorporating temporal factors.

2) **The IPUMC model** proposed a hybrid approach that combines implicit similarity derived from check-ins with users' geographical proximity. By explicitly integrating the spatial dimension into the similarity calculation, IPUMC showed a notable performance improvement over traditional methods such as Pearson correlation and cosine similarity. This contribution highlights the importance of incorporating geographical contexts to better capture users' behavioral and contextual affinities.

3) **The IUPJS model** further enriched this approach by combining the Jaccard index with a geographical influence measure based on users' initial and final check-ins. Experimental results confirmed the superiority of this approach, which generates recommendations that are both precise and contextually relevant. IUPJS thus illustrates the potential of recommendation systems to integrate multidimensional factors to better meet individual users' needs.

These contributions highlight the importance of integrating contextual factors such as geographical location, historical behaviors, and social interactions to improve POI recommendation quality. They also open several research avenues:

1) **Integration of additional contextual data**: The inclusion of semantic attributes of POIs (categories, user reviews), regional characteristics (accessibility, cultural appeal), or weather and temporal conditions could further refine recommendations.

2) **Algorithmic optimization**: Reducing the complexity of similarity calculations is essential for enabling real-time applications.

3) **Exploration of deep learning techniques**: Using machine learning models could optimize the weighting of various factors (geographical, social, behavioral) for even more tailored recommendations.

In conclusion, this thesis demonstrates that integrating spatial and behavioral dimensions into POI recommendation systems represents a major advancement in improving their relevance and accuracy. This work paves the way for smarter recommendation systems capable of providing personalized and meaningful suggestions in complex environments, such as dense urban areas or tourist destinations. It thus contributes to enhancing the user experience and addressing the challenges posed by the rise of geolocation technologies.

.

Bibliography

[1] R. J. U. m. Burke and u.-a. interaction, "Hybrid recommender systems: Survey and experiments," vol. 12, pp. 331-370, 2002.

[2] C. C. Aggarwal, "An introduction to recommender systems," in *Recommender systems: The textbook*: Springer, 2016, pp. 1-28.

[3] J. A. Konstan, J. Riedl, A. Borchers, and J. L. Herlocker, "Recommender systems: A grouplens perspective," in *Recommender Systems: Papers from the 1998 Workshop (AAAI Technical Report WS-98-08)*, 1998, pp. 60-64: AAAI Press Menlo Park.

[4] C. C. Aggarwal, *Recommender systems*. Springer, 2016.

[5] F. Ricci, L. Rokach, and B. Shapira, "Introduction to recommender systems handbook," in *Recommender systems handbook*: Springer, 2010, pp. 1-35.

[6] G. Adomavicius, A. J. I. t. o. k. Tuzhilin, and d. engineering, "Toward the next generation of recommender systems: A survey of the state-of-the-art and possible extensions," vol. 17, no. 6, pp. 734-749, 2005.

[7] X. Su and T. M. J. A. i. a. i. Khoshgoftaar, "A survey of collaborative filtering techniques," vol. 2009, no. 1, p. 421425, 2009.

[8] K. N. J. D. J. o. L. Rao and I. Technology, "Application domain and functional classification of recommender systems--a survey," vol. 28, no. 3, p. 17, 2008.

[9] M. HADJHENNI, N. DENNOUNI, Z. J. I. J. o. C. SLAMA, and D. Systems, "Toward a systematic evaluation approach of point-of-interest recommendation algorithms of a novel Smart tourism tool," vol. 15, no. 1, pp. 653-670, 2024.

[10] D. Lian, H. Wang, Z. Liu, J. Lian, E. Chen, and X. Xie, "Lightrec: A memory and search-efficient recommender system," in *Proceedings of The Web Conference 2020*, 2020, pp. 695-705.

[11] M. H. Mohamed, M. H. Khafagy, and M. H. Ibrahim, "Recommender systems challenges and solutions survey," in *2019 International Conference on Innovative Trends in Computer Engineering (ITCE)*, 2019, pp. 149-155: IEEE.

[12] S. H. Park, K. J. J. o. A. I. Kim, and H. Computing, "Collaborative filtering recommendation system based on improved Jaccard similarity," pp. 1-18, 2023.

[13] B. K. J. J. o. C. Mylavarapu and Communications, "Collaborative filtering and artificial neural network based recommendation system for advanced applications," vol. 6, no. 12, pp. 1-14, 2018.

[14] G. Behera and N. Nain, "Trade-off between memory and model-based collaborative filtering recommender system," in *Proceedings of the International Conference on Paradigms of Communication, Computing and Data Sciences: PCCDS 2021*, 2022, pp. 137-146: Springer.

[15] P. Thakkar, K. Varma, V. Ukani, S. Mankad, and S. Tanwar, "Combining user-based and item-based collaborative filtering using machine learning," in *Information and Communication Technology for Intelligent Systems*: Springer, 2019, pp. 173-180.

[16] H. Khojamli and J. J. E. S. w. A. Razmara, "Survey of similarity functions on neighborhood-based collaborative filtering," vol. 185, p. 115482, 2021.

[17] C. G. Reswara, J. Nicolas, I. M. D. Widyatama, D. David, and P. Arisaputra, "Book recommendation system using TF-IDF and cosine similarity," in *AIP Conference Proceedings*, 2024, vol. 3135, no. 1, p. 020003: AIP Publishing LLC.

[18] B. Sarwar, G. Karypis, J. Konstan, and J. Riedl, "Item-based collaborative filtering recommendation algorithms," in *Proceedings of the 10th international conference on World Wide Web*, 2001, pp. 285-295.

[19] D. Jannach and G. Adomavicius, "Recommender systems: Challenges and research opportunities," ed: Springer, 2016.

[20] Y. Koren, R. Bell, and C. J. C. Volinsky, "Matrix factorization techniques for recommender systems," vol. 42, no. 8, pp. 30-37, 2009.

[21] M. Chiny, M. Chihab, O. Bencharef, and Y. J. n. B. Chihab, "Netflix recommendation system based on TF-IDF and cosine similarity algorithms," pp. 15-20, 2022.

[22] F. O. J. I. J. o. R. Isinkaye, "Matrix factorization in recommender systems: algorithms, applications, and peculiar challenges," vol. 69, no. 9, pp. 6087-6100, 2023.

[23] J. Bobadilla, F. Ortega, A. Hernando, and A. J. K.-b. s. Gutiérrez, "Recommender systems survey," vol. 46, pp. 109-132, 2013.

[24] U. Javed, K. Shaukat, I. A. Hameed, F. Iqbal, T. M. Alam, and S. J. I. J. o. E. T. i. L. Luo, "A review of content-based and context-based recommendation systems," vol. 16, no. 3, pp. 274-306, 2021.

[25] F. Ricci, L. Rokach, and B. J. R. s. h. Shapira, "Recommender systems: Techniques, applications, and challenges," pp. 1-35, 2021.

[26] S. Khusro, Z. Ali, and I. Ullah, "Recommender systems: issues, challenges, and research opportunities," in *Information science and applications (ICISA) 2016*, 2016, pp. 1179-1189: Springer.

[27] J. B. Schafer, D. Frankowski, J. Herlocker, and S. Sen, "Collaborative filtering recommender systems," in *The adaptive web*: Springer, 2007, pp. 291-324.

[28] Y. Tai, Z. Sun, and Z. Yao, "Content-based recommendation using machine learning," in *2021 IEEE 31st International Workshop on Machine Learning for Signal Processing (MLSP)*, 2021, pp. 1-4: IEEE.

[29] T. Widiyaningtyas, I. Hidayah, and T. B. J. J. o. B. D. Adji, "User profile correlation-based similarity (UPCSim) algorithm in movie recommendation system," vol. 8, no. 1, pp. 1-21, 2021.

[30] K. W. J. N. L. E. Church, "Word2Vec," vol. 23, no. 1, pp. 155-162, 2017.

[31] S. R. Bashir, S. Raza, and V. B. J. F. I. Misic, "BERT4Loc: BERT for Location—POI recommender system," vol. 15, no. 6, p. 213, 2023.

[32] R. Kuo and H.-R. J. A. S. C. Cheng, "A content-based recommender system with consideration of repeat purchase behavior," vol. 127, p. 109361, 2022.

[33] S. J. J. o. I. Qianna and F. Systems, "Evaluation model of classroom teaching quality based on improved RVM algorithm and knowledge recommendation," vol. 40, no. 2, pp. 2457-2467, 2021.

[34] Q. Guo, "Graph-based point-of-interest recommendation on location-based social networks," 2019.

[35] A. Da'u and N. J. A. I. R. Salim, "Recommendation system based on deep learning methods: a systematic review and new directions," vol. 53, no. 4, pp. 2709-2748, 2020.

[36] D. Liang, R. G. Krishnan, M. D. Hoffman, and T. Jebara, "Variational autoencoders for collaborative filtering," in *Proceedings of the 2018 world wide web conference*, 2018, pp. 689-698.

[37] S. Liang, Z. Pan, W. Liu, J. Yin, and M. J. A. C. S. De Rijke, "A survey on variational autoencoders in recommender systems," vol. 56, no. 10, pp. 1-40, 2024.

[38] C. Gao, X. Wang, X. He, and Y. Li, "Graph neural networks for recommender system," in *Proceedings of the fifteenth ACM international conference on web search and data mining*, 2022, pp. 1623-1625.

[39] H. WEI, K. LI, X. HE, Z. J. G. TIAN, and I. S. o. W. University, "Integrating spatial relationship into a matrix factorization model for POI recommendation," vol. 46, no. 5, pp. 681-690, 2021.

[40] Y.-M. Li, C.-T. Wu, and C.-Y. J. D. s. s. Lai, "A social recommender mechanism for e-commerce: Combining similarity, trust, and relationship," vol. 55, no. 3, pp. 740-752, 2013.

[41] H. I. Pohan, H. L. H. S. Warnars, B. Soewito, and F. L. Gaol, "Recommender system using transformer model: A systematic literature review," in *2022 1st International Conference on Information System & Information Technology (ICISIT)*, 2022, pp. 376-381: IEEE.

[42] J. Chang *et al.*, "Sequential recommendation with graph neural networks," in *Proceedings of the 44th international ACM SIGIR conference on research and development in information retrieval*, 2021, pp. 378-387.

[43] H. Chen *et al.*, "Denoising self-attentive sequential recommendation," in *Proceedings of the 16th ACM conference on recommender systems*, 2022, pp. 92-101.

[44] F. Sun *et al.*, "BERT4Rec: Sequential recommendation with bidirectional encoder representations from transformer," in *Proceedings of the 28th ACM international conference on information and knowledge management*, 2019, pp. 1441-1450.

[45] H. Wang, X. Shen, M. Tu, Y. Zhuang, Z. J. I. A. T. o. A. Liu, Speech,, and L. Processing, "Improved transformer with multi-head dense collaboration," vol. 30, pp. 2754-2767, 2022.

[46] Y. Deldjoo *et al.*, "Recommendation with generative models," arXiv preprint arXiv:2409.15173. 2024

[47] T. Bai *et al.*, "GPR-OPT: A Practical Gaussian optimization criterion for implicit recommender systems," vol. 61, no. 1, p. 103525, 2024.

[48] I. Saifudin and T. J. I. A. Widiyaningtyas, "Systematic literature review on recommender system: Approach, problem, evaluation techniques, datasets," vol. 12, pp. 19827-19847, 2024.

[49] P. Zhang *et al.*, "TransGNN: Harnessing the collaborative power of transformers and graph neural networks for recommender systems," in *Proceedings of the 47th International ACM SIGIR Conference on Research and Development in Information Retrieval*, 2024, pp. 1285-1295.

[50] B. Jin, Y. Zhang, Q. Zhu, and J. Han, "Heterformer: Transformer-based deep node representation learning on heterogeneous text-rich networks," in *Proceedings of the 29th ACM SIGKDD conference on knowledge discovery and data mining*, 2023, pp. 1020-1031.

[51] X. Wang, Q. Li, D. Yu, Q. Li, and G. J. A. T. o. I. S. Xu, "Counterfactual explanation for fairness in recommendation," vol. 42, no. 4, pp. 1-30, 2024.

[52] W. Juan, L. Yue-xin, and W. Chun-ying, "Survey of recommendation based on collaborative filtering," in *Journal of Physics: Conference Series*, 2019, vol. 1314, no. 1, p. 012078: IOP Publishing.

[53] P. B. Thorat, R. M. Goudar, and S. J. I. J. o. C. A. Barve, "Survey on collaborative filtering, content-based filtering and hybrid recommendation system," vol. 110, no. 4, 2015.

[54] M. H. Chitapulla, S. Assi, W. Bajnaid, M. Jayabalan, and D. Al-Jumeily, "Evaluation of Hybrid Recommendation System and Machine Learning Algorithms for E-Commerce Platform," in *International Conference on Mechatronics and Intelligent Robotics*, 2023, pp. 393-405: Springer.

[55] A. B. Suhaim and J. J. I. A. Berri, "Context-aware recommender systems for social networks: review, challenges and opportunities," vol. 9, pp. 57440-57463, 2021.

[56] F. M. Harper and J. A. J. A. t. o. i. i. s. Konstan, "The movielens datasets: History and context," vol. 5, no. 4, pp. 1-19, 2015.

[57] E. Zangerle and C. J. A. c. s. Bauer, "Evaluating recommender systems: survey and framework," vol. 55, no. 8, pp. 1-38, 2022.

[58] K. C. Jena, S. Mishra, S. Sahoo, and B. K. Mishra, "Principles, techniques and evaluation of recommendation systems," in *2017 International Conference on Inventive Systems and Control (ICISC)*, 2017, pp. 1-6: IEEE.

[59] H. Jiao, F. Mo, and H. Yamana, "Evaluation of POI recommendation system beyond accuracy: Diversity explainability and computation cost," in *Proceedings of 18th Japan data engineering and information management (DEIM) forum*, 2020.

[60] J. L. Herlocker, J. A. Konstan, L. G. Terveen, and J. T. J. A. T. o. I. S. Riedl, "Evaluating collaborative filtering recommender systems," vol. 22, no. 1, pp. 5-53, 2004.

[61] P. Cremonesi, Y. Koren, and R. Turrin, "Performance of recommender algorithms on top-n recommendation tasks," in *Proceedings of the fourth ACM conference on Recommender systems*, 2010, pp. 39-46.

[62] J. Basilico and T. Hofmann, "Unifying collaborative and content-based filtering," in *Proceedings of the twenty-first international conference on Machine learning*, 2004, p. 9.

[63] B. Mobasher, "Data mining for web personalization," in *The adaptive web: Methods and strategies of web personalization*: Springer, 2007, pp. 90-135.

[64] R. Kohavi, R. J. E. o. m. l. Longbotham, and d. mining, "Online controlled experiments and A/B tests," pp. 1-11, 2015.

[65] J. B. Schafer, J. Konstan, and J. Riedl, "Recommender systems in e-commerce," in *Proceedings of the 1st ACM conference on Electronic commerce*, 1999, pp. 158-166.

[66] T. J. A. T. o. I. S. Hofmann, "Latent semantic models for collaborative filtering," vol. 22, no. 1, pp. 89-115, 2004.

[67] P. Resnick and H. R. J. C. o. t. A. Varian, "Recommender systems," vol. 40, no. 3, pp. 56-58, 1997.

[68] G. Shani and A. J. R. S. H. Gunawardana, "Evaluating recommender systems," pp. 257-297, 2015.

[69] C. M. Bishop and N. M. Nasrabadi, *Pattern recognition and machine learning* (no. 4). Springer, 2006.

[70] M. Kuhn and K. Johnson, *Applied predictive modeling.* Springer, 2013.

[71] J. J. Friedman, "The elements of statistical learning: Data mining, inference, and prediction," 2009.

[72] C. D. Manning, *An introduction to information retrieval.* 2009.

[73] M. Jalili, S. Ahmadian, M. Izadi, P. Moradi, and M. J. I. a. Salehi, "Evaluating collaborative filtering recommender algorithms: a survey," vol. 6, pp. 74003-74024, 2018.

[74] R. Zhang, H. Bao, H. Sun, Y. Wang, and X. J. F. o. C. S. Liu, "Recommender systems based on ranking performance optimization," vol. 10, pp. 270-280, 2016.

[75] M. HADJHENNI, N. DENNOUNI, and Z. J. N. SLAMA, "Orchestration of POIs ubiquitous contexts: a review of recommendation systems based on matrix factorization model," p. 55, 2023.

[76] X. Wei, Y. Qian, C. Sun, J. Sun, and Y. J. G. Liu, "A survey of location-based social networks: problems, methods, and future research directions," vol. 26, no. 1, pp. 159-199, 2022.

[77] W. Chen *et al.*, "Building and exploiting spatial–temporal knowledge graph for next POI recommendation," vol. 258, p. 109951, 2022.

[78] S. Kulkarni and S. F. J. C. S. R. Rodd, "Context Aware Recommendation Systems: A review of the state of the art techniques," vol. 37, p. 100255, 2020.

[79] Q. Liu *et al.*, "Multimodal recommender systems: A survey," vol. 57, no. 2, pp. 1-17, 2024.

[80] N. Idrissi, A. J. S. N. A. Zellou, and Mining, "A systematic literature review of sparsity issues in recommender systems," vol. 10, no. 1, p. 15, 2020.

[81] X. Xiong *et al.*, "Where to go: An effective point-of-interest recommendation framework for heterogeneous social networks," vol. 373, pp. 56-69, 2020.

[82] L. Shi, S. Li, X. Ding, and Z. J. E. S. w. A. Bu, "Selection bias mitigation in recommender system using uninteresting items based on temporal visibility," vol. 213, p. 118932, 2023.

[83] H. Yuan and A. A. J. I. a. Hernandez, "User cold start problem in recommendation systems: A systematic review," vol. 11, pp. 136958-136977, 2023.

[84] J. Zeng, H. Tang, Y. Zhao, M. Gao, J. J. M. N. Wen, and Applications, "PR-RCUC: A POI recommendation model using region-based collaborative filtering and user-based mobile context," vol. 26, no. 6, pp. 2434-2444, 2021.

[85] W. Wang *et al.*, "User-context collaboration and tensor factorization for gnn-based social recommendation," vol. 10, no. 6, pp. 3320-3330, 2023.

[86] X. Li, L. Sun, M. Ling, and Y. J. N. Peng, "A survey of graph neural network based recommendation in social networks," vol. 549, p. 126441, 2023.

[87] N. L. Ho and K. H. Lim, "Poibert: A transformer-based model for the tour recommendation problem," in *2022 IEEE International Conference on Big Data (Big Data)*, 2022, pp. 5925-5933: IEEE.

[88] S. Halder, K. H. Lim, J. Chan, and X. Zhang, "Transformer-based multi-task learning for queuing time aware next poi recommendation," in *Pacific-Asia conference on knowledge discovery and data mining*, 2021, pp. 510-523: Springer.

[89] K. Vaswani, Y. Agrawal, and V. Alluri, "Multimodal fusion based attentive networks for sequential music recommendation," in *2021 IEEE seventh international conference on multimedia big data (BigMM)*, 2021, pp. 25-32: IEEE.

[90] N. Dennouni, "Orchestration des activités d'apprentissage mobile," Université de Lille 1-Sciences et Technologies, 2016.

[91] R. Gao *et al.*, "Exploiting geo-social correlations to improve pairwise ranking for point-of-interest recommendation," vol. 15, no. 7, pp. 180-201, 2018.

[92] C. Cheng, H. Yang, I. King, and M. Lyu, "Fused matrix factorization with geographical and social influence in location-based social networks," in *Proceedings of the AAAI conference on artificial intelligence*, 2012, vol. 26, no. 1, pp. 17-23.

[93] D. Lian, C. Zhao, X. Xie, G. Sun, E. Chen, and Y. Rui, "GeoMF: joint geographical modeling and matrix factorization for point-of-interest recommendation," in *Proceedings of the 20th ACM SIGKDD international conference on Knowledge discovery and data mining*, 2014, pp. 831-840.

[94] J.-D. Zhang and 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*, 2015, pp. 443-452.

[95] Y. Liu, T.-A. N. Pham, G. Cong, and Q. Yuan, "An experimental evaluation of point-of-interest recommendation in location-based social networks," 2017.

[96] S. Zhao, T. Zhao, I. King, and M. R. Lyu, "Geo-teaser: Geo-temporal sequential embedding rank for point-of-interest recommendation," in *Proceedings of the 26th international conference on world wide web companion*, 2017, pp. 153-162.

[97] J. Wang, L. Zhang, Z. Sun, and Y.-S. Ong, "Meta-learning enhanced next POI recommendation by leveraging check-ins from auxiliary cities," in *Pacific-Asia Conference on Knowledge Discovery and Data Mining*, 2023, pp. 322-334: Springer.

[98] J. Chen, W. Zhang, P. Zhang, P. Ying, K. Niu, and M. J. C. Zou, "Exploiting spatial and temporal for point of interest recommendation," vol. 2018, no. 1, p. 6928605, 2018.

[99] R. Ding, Z. Chen, and X. J. I. A. Li, "Spatial-temporal distance metric embedding for time-specific POI recommendation," vol. 6, pp. 67035-67045, 2018.

[100] H. Gao, J. Tang, X. Hu, and H. Liu, "Content-aware point of interest recommendation on location-based social networks," in *Proceedings of the AAAI conference on artificial intelligence*, 2015, vol. 29, no. 1.

[101] F. Huang, S. Qiao, J. Peng, B. Guo, and N. J. I. T. o. E. T. i. C. Han, "STPR: a personalized next point-of-interest recommendation model with spatio-temporal effects based on purpose ranking," vol. 9, no. 2, pp. 994-1005, 2019.

[102] X. Jiao, Y. Xiao, W. Zheng, H. Wang, and Y. Jin, "R2SIGTP: A novel real-time recommendation system with integration of geography and temporal preference for next point-of-interest," in *The world wide web conference*, 2019, pp. 3560-3563.

[103] W. Wang, J. Chen, J. Wang, J. Chen, J. Liu, and Z. J. I. T. o. I. I. Gong, "Trust-enhanced collaborative filtering for personalized point of interests recommendation," vol. 16, no. 9, pp. 6124-6132, 2019.

[104] Y. Zhang *et al.*, "Personalized geographical influence modeling for POI recommendation," vol. 35, no. 5, pp. 18-27, 2020.

[105] D. Bettache, D. J. I. J. o. C. Nassim, and D. Systems, "Toward a New Similarity Measure Based on Combining Tourist Check-ins and Their Trip Path for a Point-Of-Interest Recommendations in a LBSN," vol. 17, no. 1, pp. 1-13, 2024.

[106] S. Qaiser and R. J. I. J. o. C. A. Ali, "Text mining: use of TF-IDF to examine the relevance of words to documents," vol. 181, no. 1, pp. 25-29, 2018.

[107] J. Ni, Y. Cai, G. Tang, and Y. J. A. S. Xie, "Collaborative filtering recommendation algorithm based on TF-IDF and user characteristics," vol. 11, no. 20, p. 9554, 2021.

[108] P. Dauni, M. Firdaus, R. Asfariani, M. Saputra, A. Hidayat, and W. Zulfikar, "Implementation of Haversine formula for school location tracking," in *Journal of Physics: Conference Series*, 2019, vol. 1402, no. 7, p. 077028: IOP Publishing.

[109] R. Purnomo, T. D. Putra, H. Kusmara, W. Priatna, and F. J. J. Mukharom, "Haversine formula to find the nearest petshop," vol. 9, no. 3, pp. 2205-2221, 2022.

[110] D. yang. Foursquare Dataset. dingqi yang's homepage, available online," https://sites.google.com/site/yangdingqi/home, accessed on 1 Feb 2024. [Online].

[111] A. A. Amer, H. I. Abdalla, and L. J. K.-B. S. Nguyen, "Enhancing recommendation systems performance using highly-effective similarity measures," vol. 217, p. 106842, 2021.

[112] A. Almu and Z. Bello, "An experimental study on the accuracy and efficiency of some similarity measures for collaborative filtering recommender systems," 2021.

[113] F. J. J. o. K. S. U.-C. Fkih and I. Sciences, "Similarity measures for Collaborative Filtering-based Recommender Systems: Review and experimental comparison," vol. 34, no. 9, pp. 7645-7669, 2022.

[114] D. Bettache, Dennouni, N. and Harbouche, A, "Improving POI Recommendation through Collaborative Filtering by incorporating the Geographical Factor into the Similarity Calculation," *Engineering, Technology & Applied Science Research,* vol. 15, 3 Jun. 2025.

[115] D. Bettache, Dennouni, N. and Harbouche, A, "Enhancing Points of Interest Recommendation by Integrating Users' Proximity into the Calculation of their Similarities," *Indonesian Journal of Electrical Engineering and Computer Science,* no. (Accepted), 2025.