CONTEXT-AWARE NEXT POINT-OF-INTEREST RECOMMENDATION WITH UNCERTAIN CHECK-INS

ZHANG LU

SCHOOL OF COMPUTER SCIENCE AND ENGINEERING

2022

CONTEXT-AWARE NEXT POINT-OF-INTEREST RECOMMENDATION WITH UNCERTAIN CHECK-INS

ZHANG LU

School of Computer Science and Engineering

A thesis submitted to the Nanyang Technological University
in partial fulfilment of the requirement for the degree of
Doctor of Philosophy (Ph.D)

2022

Statement of Originality

I hereby certify that the work embodied in this thesis is the result of original research, is free of plagiarised materials, and has not been submitted for a higher degree to any other University or Institution.

9/3/2022 Zhang Lu

. . . .

Date Zhang Lu

Supervisor Declaration Statement

I have reviewed the content and presentation style of this thesis and declare it is free of plagiarism and of sufficient grammatical clarity to be examined. To the best of my knowledge, the research and writing are those of the candidate except as acknowledged in the Author Attribution Statement. I confirm that the investigations were conducted in accord with the ethics policies and integrity standards of Nanyang Technological University and that the research data are presented honestly and without prejudice.

9/3/2022

. . . .
Date

. . . .
Assoc Prof Jie Zhang

Authorship Attribution Statement

This thesis contains material from 3 papers published in the following peer-reviewed journals and conference in which I am listed as an author.

Chapter 3 is published as **Lu Zhang**, Zhu Sun, Jie Zhang, Horst Kloeden and Felix Klanner, "Modeling hierarchical category transition for next POI recommendation with uncertain check-ins." Information Sciences 515 (2020): 169-190.
DOI: https://doi.org/10.1016/j.ins.2019.12.006

The contributions of the co-authors are as follows:

- I designed the model, conducted the data analysis and experiments, and prepared the manuscript draft. Dr. Zhu Sun discussed with me about how to properly present the key idea of the manuscript and revise it.
- Prof. Jie Zhang provided the initial project direction and revised the manuscript.
- Dr. Horst Kloeden and Dr. Felix Klanner are the project founding providers.

Chapter 4 is published as **Lu Zhang**, Zhu Sun, Jie Zhang, Yu Lei, Chen Li, Ziqing Wu, Horst Kloeden and Felix Klanner. "An Interactive Multi-Task Learning Framework for Next POI Recommendation with Uncertain Check-ins." In Proceedings of the 29th International Joint Conference on Artificial Intelligence, pp. 3551-3557, 2020. DOI: https://doi.org/10.24963/ijcai.2020/491.

The contributions of the co-authors are as follows:

- Prof. Jie Zhang and Dr. Zhu Sun provided the initial project direction.
- I designed the model, conducted the experiments and prepared the manuscript. Dr. Zhu Sun discussed with me about the framework of the model.
- Prof. Jie Zhang and Dr. Zhu Sun revised the manuscript.
- Yu Lei, Chen Li and Ziqing Wu assisted in the data construction, data analysis and experiment conduction.
- Dr. Horst Kloeden and Dr. Felix Klanner are the project founding providers.

Chapter 5 is published as **Lu Zhang**, Zhu Sun, Jie Zhang, Yiwen Wu and Yunwen Xia. "Conversation-based Adaptive Relational Translation Method for Next POI Recommendation with Uncertain Check-ins." IEEE Transactions on Neural Networks and Learning Systems, 2022. DOI: 10.1109/tnnls.2022.3146443

The contributions of the co-authors are as follows:

- Prof. Jie Zhang and Dr. Zhu Sun provided the initial project direction.
- I designed the model, conducted the experiments and prepared the manuscript draft.
- Prof. Jie Zhang and Dr. Zhu Sun revised the manuscript.
- Yiwen Wu and Yunwen Xia assisted in the experiment conduction.

9/3/2022	Zhang Lu
. . . .	. . . .
Date	Zhang Lu

Acknowledgments

I would like to express my sincere gratitude towards my supervisor Prof. Jie Zhang, for his patient guidance, encouragement, and great support during my Ph.D. journey. His immense knowledge and insightful advice not only help me learn how to do research but also benefit my future career development. I feel very grateful to my co-author Dr. Zhu Sun for her constant support and elaborative guidance during the past three years. Many thanks for her shared generous experience in research. I would like to thank Prof. Gao Cong and Prof. Hong Xu for being my Thesis Advisory Committee members.

I would also like to appreciate my group members for their friendly help. They include Zhiguang Cao, Wen Song, Ziqing Wu, Yiwen Wu, Yunwen Xia, Yu Lei, Chen Li, Cong Zhang, Yaoxin Wu, Xiao Shao. I also want to thank my colleagues and administrators at BMW-NTU Research Lab: Chan Jyh Huah, Jagadheesan Vignesh and Maggie Lee Suh Chya. I am truly thankful for the support provided by BMW group members, Horst Kloeden and Felix Klanner.

Last but not least, I would like to sincerely thank my parents and my husband for their constant support, encouragement and love all the time.

Contents

Acknowledgments v
Summary ix
List of Figures xi
List of Tables 1

1 Introduction **2**
1.1 Research Scope 2
1.2 Research Challenges 6
1.3 Research Approaches 8
1.4 Research Contributions 10
1.5 Dissertation Organization 12

2 Literature Review **13**
2.1 Context Features for Next POI Recommendation 13
2.1.1 Categorical Context 14
2.1.2 Spatial Context 14
2.1.3 Temporal Context 15
2.1.4 Sequential Context 16
2.2 Hierarchical Category Transition based Recommendation 16
2.2.1 Latent Factor Model based Approaches 17
2.2.2 Representation Learning Model based Approaches 18
2.3 Multi-task Learning based Recommendation 19
2.3.1 Activity and Location aware Recommendation 19
2.3.2 Multi-Task Learning for Recommendation 20
2.4 Conversation based Recommendation 22

2.4.1 Static Next POI Recommendation 22
2.4.2 Conversational Recommender System 23
2.5 Summary 24

3 HCT: Hierarchical Category Transition Method 26
3.1 Data Construction and Analysis 27
3.1.1 Data Construction 27
3.1.2 Empirical Analysis 31
3.2 Problem Formulation 34
3.3 The HCT Framework 35
3.3.1 Data Module 35
3.3.2 Training Module 37
3.3.3 Prediction Module 39
3.3.4 Model Optimization and Complexity Analysis 42
3.4 Experiments and Analysis 43
3.4.1 Experimental Setup 43
3.4.2 Results and Analysis 46
3.5 Summary 53

4 iMTL: Interactive Multi-task Learning Method 54
4.1 Data Description and Analysis 55
4.2 Problem Formulation 57
4.3 The iMTL Framework 58
4.3.1 Temporal-aware Activity Encoder 58
4.3.2 Spatial-aware Location Preference Encoder 59
4.3.3 Task-specific Decoder 60
4.3.4 Learning Strategy and Complexity Analysis 61
4.4 Experiments and Analysis 62
4.4.1 Experimental Setup 63
4.4.2 Performance Comparison (RQ1) 65
4.4.3 Detailed Study of iMTL (RQ2) 67
4.4.4 Parameter Sensitivity Analysis (RQ3) 67
4.5 Summary 68

5 CART: Conversation-based Adaptive Relational Translation Method 69
5.1 The CART Framework 70
5.1.1 Recommender Module 71
5.1.2 Conversation Module 74
5.1.3 Connection of Recommender and Conversation Modules 77
5.1.4 Discussion on the Issue of Uncertain Check-ins 77
5.1.5 Model Optimization 78
5.2 Experiments and Analysis 79
5.2.1 Experimental Setup 80
5.2.2 Performance Comparison for CRS Methods (RQ1) 84
5.2.3 Performance Comparison for Static Methods (RQ2) 85
5.2.4 Study on Quality of User Response (RQ3) 89
5.2.5 Ablation Study (RQ4) 91
5.2.6 Hyper-parameter Analysis (RQ5) 93
5.3 Summary 93

6 Conclusions and Future Work 94
6.1 Conclusions 94
6.2 Future Work 96

List of Publications 98

List of Submissions 99

References 100

Summary

The rapid development of next point-of-interest (POI) recommendation benefits from a large number of check-ins shared by users in location-based social networks (LBSNs), such as Foursquare and Yelp, which helps users explore their surroundings. Accordingly, most existing studies assume that such check-ins reflect users' exact visits, i.e., certain check-ins. In reality, due to the privacy concern or the bias of the indoor navigation of GPS, users may leave some uncertain check-ins at a collective POI, e.g., a shopping mall containing multiple individual POIs. Such uncertain check-ins bring the challenges of learning user preference and complete transition patterns, which are rarely investigated by the existing next POI recommendation studies. Therefore, in this dissertation, we focus on a new research problem that aims to recommend next individual POIs with uncertain check-ins, and develop a series of effective algorithms by exploiting context information mined from users' historical check-in behaviors to resolve this problem.

In the next POI recommendation scenario, exploiting users' preference transitions over the categories of POIs, i.e., category transition, is of significance in improving recommendation accuracy. Moreover, POIs are typically organized by a category hierarchy (CH). As a result, CH can better describe the correlations between POIs (categories) and categories, which shows potential in easing the cold start issue in the recommendation task. Existing studies either exploit the category transition or CH to improve the recommendation performance. However, they lack consideration of both effects in a unified manner. In order to take advantage of their joint effects for better recommendation under the uncertain check-in scenario, we first propose a HCT framework, which exploits hierarchical category transitions for better multi-granularity user preference transition learning. In this way, HCT predicts users' preferred categories inside collective POIs. As bounded to specific categories, HCT further adopts hierarchical dependencies in CH

to capture the semantic relatedness of POIs, thus easing the cold start issue. Empirical studies show the superiority of HCT against state-of-the-art algorithms.

Despite the pioneer effort HCT can address the challenge of uncertain check-ins to some extent, it fails to well characterize users' underlying activities over uncertain check-ins and model the interplay between sequential activities and locations by considering the spatiotemporal context. Therefore, we devise an interactive multi-task learning framework – iMTL, which introduces: a temporal-aware activity encoder equipped with fuzzy characterization to unveil the latent activity transition patterns; a spatial-aware location preference encoder to capture the latent location transition patterns; and a task-specific decoder to adopt the learned latent transition patterns and enhance both activity and location prediction tasks in an interactive manner. Extensive experiments on real-world datasets demonstrate that iMTL consistently outperforms the state of the arts.

Although the proposed HCT and iMTL have shown the capability of easing the issue of uncertain check-ins, they suffer from the inability of recommending accurate POIs when they fail to predict user's accurate activity (i.e., immediate preference). In fact, most existing next POI recommenders (i.e., static methods) face the inherent limitations of capturing users' accurate immediate preferences and adapting to their feedback regarding the recommendation results. The development of conversational recommender system (CRS) techniques brings great potential in resolving the limitations of such static recommenders. We further propose a conversation-based adaptive relational translation (CART) approach for next POI recommendation over uncertain check-ins. It is equipped with recommender and conversation modules to interactively acquire a user's immediate preference and make dynamic recommendations. Specifically, the recommender built upon the adaptive relational translation method performs location prediction via modeling both sequential behaviors and the immediate preference received from conversations; and the conversation module aims to achieve successful recommendations in fewer conversation turns by learning a conversational strategy, whereby the recommender can be updated via the user's feedback. Experiments on real-world datasets show the superiority of our proposed CART over state-of-the-art algorithms.

To sum up, in this dissertation, we propose a series of recommendation approaches by exploiting context information mined from users' check-in behaviors for more accurate next POI recommendations under the uncertain check-in scenario.

List of Figures

1.1 An example of Bob's certain and uncertain check-ins. 3
1.2 An example of a two-layer category hierarchy for restaurants. 4
1.3 An illustration of conversation-based next POI recommendation. 10

3.1 A running example to illustrate Alice's check-in sequences. 28
3.2 All individual POIs. 31
3.3 Examples of collective POIs. 31
3.4 Distribution of sequences. 32
3.5 Distribution of check-ins. 33
3.6 Distribution of check-ins. 33
3.7 An example to illustrate the overview framework of the proposed HCT. . 36
3.8 The illustration of modeling the hierarchical dependencies and category transition. 38
3.9 Performance comparison for top-K POI recommendation. 47
3.10 Performance comparison w.r.t. category correctness on the three datasets. 48
3.11 Performance comparison for the Top-K_{poi} individual POI recommendation. 49
3.12 The effect of embedding size for HCT. 50
3.13 The effect of window size for HCT. 51
3.14 The effect of α_j for HCT. 52
3.15 The effect of data sparsity for HCT. 52

4.1 Three observations obtained from the data analysis. 56
4.2 The architecture of the proposed iMTL. 58
4.3 Performance comparison for variants of iMTL for activity prediction. . . 66
4.4 Performance comparison for variants of iMTL for POI prediction. 66

4.5 Parameter sensitivity analysis on POI prediction task. 67

5.1 The framework of CART, which is composed of recommender and conversation modules. 72
5.2 Success Rate* (SR*@n) of CRS methods at each conversation turn. . . . 87
5.3 Performance comparison between static recommendation baselines and the proposed CART. 88
5.4 Results of parameter sensitivity for CART. 92

List of Tables

3.1 The statistics of three original POI datasets. 29
3.2 A running example of mapping the original check-ins to newly-constructed check-ins. 30
3.3 The statistics of three newly-constructed POI datasets. 30
3.4 Notations for HCT. 34
3.5 Update Rules for HCT. 42

4.1 The statistics of three newly-constructed POI datasets for iMTL. 56
4.2 Performance comparison of activity and location tasks on the three datasets. 65

5.1 Notations for CART 71
5.2 Statistics of the four datasets for CART. 80
5.3 Performance (represented by means and standard deviations) of all methods on the four datasets measured by SR@10 and AT, where the best performance is boldfaced; and the runner up is underlined. 86
5.4 Performance comparison of the CART and CART-*rand* with fixed maximum turn and random turn, respectively. 89
5.5 Performance (represented by means and standard deviations) of different variants of CART w.r.t. SR and AT across the four datasets, where the best performance is boldfaced; and the runner up is underlined. 90

Chapter 1

Introduction

1.1 Research Scope

In recent years, the development of location-based social networks, such as Foursquare[1] and Yelp[2], has greatly promoted active research in the area of point-of-interest (POI) recommendation [1, 2] and thus benefits both personal assistant service and businesses (POI holders). One prominent research line is the next POI recommendation (a.k.a sequential POI recommendation) where the goal is to recommend locations[3], such as restaurants, that users might be interested in at the next timestamp [3, 4]. Existing studies, however, are mainly based on the commonly utilized check-in datasets (e.g. Foursquare) and assume that such check-ins reflect users' exact POI visits [5–8], i.e., certain check-ins. In reality, due to privacy concerns or the bias of the indoor navigation of GPS, users do not always disclose some specific POI visits, especially in the gathering places of numerous individual POIs, such as a shopping mall (namely collective POI) that contains multiple individual POIs. Hence, they may leave uncertain check-ins at a collective POI.

As illustrated in Fig. 1.1, Bob checked in at *individual POIs* l_1 (e.g., office) at time t_1 and l_2 (e.g., gym) at time t_2 successively, namely *certain check-in*. He then visited the individual POI l_6 inside the Building l_5, which is defined as *collective POI* containing multiple individual POIs (e.g., l_6, l_7, l_8). Most existing sequential POI recommendation studies are built upon Bob's certain check-ins: $l_1 \rightarrow l_2 \rightarrow l_6$. However, Bob may

[1] https://foursquare.com/
[2] https://www.yelp.com/
[3] 'POI' and 'location' are interchangeably used in this dissertation.

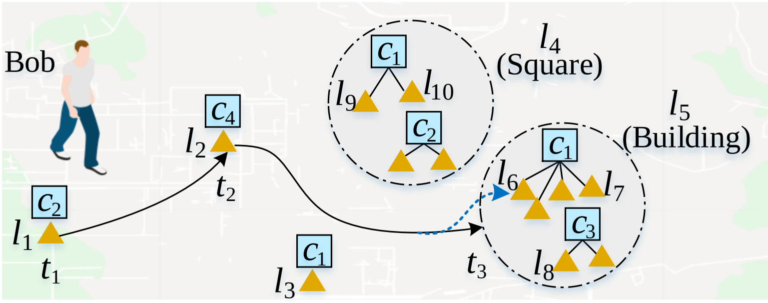


Fig. 1.1: An example of Bob's certain and uncertain check-ins, where yellow triangles are individual POIs (e.g., l_1, l_2); dotted circles are collective POIs (e.g., l_4, l_5); blue squares are categories (e.g., c_1, c_2) that POIs belong to; t_i is the particular check-in time.

leave a rough footprint, e.g., l_5, instead of the precise POI l_6. The accessible record thus will become: $l_1 \rightarrow l_2 \rightarrow l_5$, where the *uncertain check-in* at collective POI l_5 is involved. Based on such records, existing POI recommenders may recommend l_5 at t_3 to Bob instead of l_6 after he visited l_2 at t_2, because POIs l_6, l_7, l_8 are unobserved in his historical records (i.e., cold start POIs).

The above example shows the presence of users' uncertain check-ins at collective POIs in reality. It meanwhile emphasizes the necessity of investigation on a new research problem, that is, recommending next precise individual POIs (either outside or inside collective POIs), e.g., l_2, l_6, to users over uncertain check-ins. This, however, is non-trivial and fronted with challenges: (1) the difficulty to learn complete and accurate users' check-in transition patterns with these uncertain check-ins at collective POIs; and (2) the aggravation on cold start issue due to the uncertain check-ins at collective POIs. To address the above challenges, in this dissertation, we focus on investigating how to exploit context information originated from users' historical check-in behaviors to achieve more accurate next POI recommendations under the uncertain check-in scenario.

In the next POI recommendation, mining insights of users' preference transitions among locations, i.e., category transition, is significant in improving recommendation accuracy. For example, Bob went to the gym (i.e., category/activity) and checked in at a specific location l_2 according to his activity. When his next preferred category c_1 is predicted (i.e., $c_4 \rightarrow c_1$), the POI candidate space is narrowed down by the predicted category. These category-aware next POI recommenders have been developed, which model the category transition and derive a list POIs by considering the category

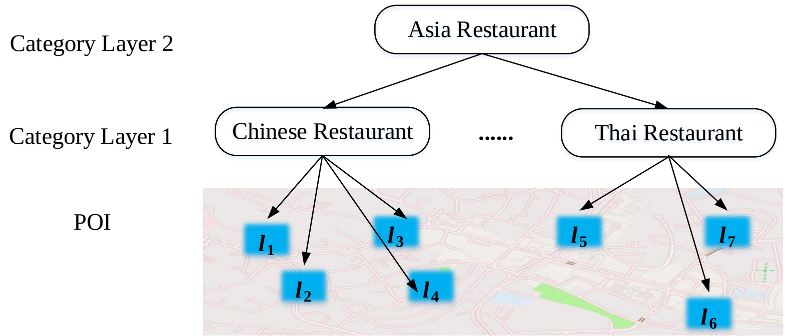


Fig. 1.2: An example of a two-layer category hierarchy for restaurants, where $l_1, \dots, l_7$ denote different restaurants; $l_1 - l_4$ are associated with the first-layer category 'Chinese Restaurant'; $l_5 - l_7$ belong to the first-layer category 'Thai Restaurant'; both 'Chinese Restaurant' and 'Thai Restaurant' at the first layer are associated with the second-layer category 'Asia Restaurant'.

preference and spatiotemporal context [1, 9, 10]. As bounded to specific categories, recommendations can be more accurate in comparison with solutions that directly model users' transition patterns among locations. Besides, as illustrated in Fig. 1.2, POIs are typically organized by the category hierarchies (CHs). The dependencies[4] between POIs (categories) and categories at different layers of CHs can help better define semantic relations of POIs from low (leaf) to high (root) levels [11, 12], thus assisting in more accurate recommendation. Many recent studies show that CHs are effective in resolving the cold start issue [13–15]. Typical examples for CHs include on-line products hierarchy (e.g., Amazon web store [14]), video hierarchy (e.g, Micro-video [16, 17]), and so forth. As there are different category transitions over different layers of CHs, it is, therefore, of great potential in exploiting both category transition and category hierarchy in a unified manner to comprehensively model user preference transition in different granularity.

Recently, multi-task learning (MTL) [18] has drawn much attention in natural language processing [19] and speech recognition [20], with recommendation system being no exception [7, 21–23]. It is capable of learning multiple related tasks simultaneously and improving model generalization by leveraging potential correlations and common features among related tasks. Recommendation systems intuitively refer to multiple related tasks,

[4]Based on the category hierarchy, we can obtain two types of dependencies: (1) the dependency between a POI and its direct (bottom) category; and (2) the dependency between two categories at any two consecutive layers of the hierarchy. For simplicity, we shorten it as: the dependencies between POIs (categories) and categories.

such as rating prediction, item recommendation, category recommendation and recommendation explanation. As a result, exploiting related tasks into the MTL framework can not only augment the single task learning but also enhance the recommendation performance. Recent location-based MTL recommenders [7, 24] incorporate multiple related tasks, such as activity and POI prediction tasks, to improve the recommendation accuracy. However, under the uncertain check-in scenario, they are insufficient in modeling rich context information (e.g., spatiotemporal and sequential regularity) of users' check-in trajectories for activity and POI predictions, as they fail to capture the underlying activity over uncertain check-ins and explore the interplay between different tasks. Hence, how to model the context information under the uncertain check-in scenario by taking advantage of the MTL framework is of significance for a more accurate next POI recommendation.

Intuitively, most existing next POI recommenders (i.e., static methods) suffer from the following inherent limitations: (1) the difficulty of capturing a user's immediate preference since user preference often drifts with time; and (2) the difficulty of updating the recommender with user's feedback regarding the recommendation results. The development of conversational recommender system (CRS) techniques [25–29] brings great potential in resolving the limitations of these static recommenders, since they allow the recommender to acquire a user's immediate preference and achieve high-quality recommendations via real-time conversations. The recent studies [30, 31] have demonstrated the efficacy of multi-round setting of CRS for more accurate item recommendations. Note that the multi-round CRS refers to that the system interacts with a user and makes recommendation multiple times until the user accepts the recommendation or chooses to quit. To this end, it is of great potential to build a location-aware CRS by integrating the merits of both multi-round CRS and context-aware recommender for an enhanced next POI recommendation over uncertain check-ins.

To summarize, in this dissertation, we devote to developing a series of novel recommendation approaches by exploiting various context information to achieve more accurate next POI recommendations with uncertain check-ins.

1.2 Research Challenges

According to the research scope illustrated in Section 1.1, three recommendation problems are defined and investigated in this dissertation, i.e., hierarchical category transition based recommendation, interactive multi-task learning based recommendation, and conversation-based recommendation. In this section, we will respectively demonstrate their research challenges.

Hierarchical Category Transition based Recommendation. A user's preference transition over the categories of POIs (i.e., category transition) is capable of reflecting the insights of his transition pattern on locations [1, 9, 10]. By studying the patterns of user preference transition over categories of certain check-ins, we are able to infer possible categories of uncertain check-ins. As bounded to specific categories, recommendations can be more accurate in comparison with conventional solutions that directly model users' transition patterns over individual POIs [11]. Some studies show that category hierarchies (CHs) can help ease the cold start issue. As an example in Fig. 1.1, individual POIs, e.g., l_6 and l_7, inside a collective POI l_5 are all cold start POIs. Based on the CHs, we can obtain their corresponding category representation from certain check-ins, which enables the recommender to infer a user's preferred category within a collective POI. In addition to cold start POIs, the presence of uncertain check-ins may also result in cold start categories at a lower layer, for example, the category c_3 is unobserved in the certain check-ins. Hence, we argue that 1) the category transitions over different layers of CHs would help capture users' preference transitions in different granularity, and 2) the dependencies between POIs (categories) and categories at different layers of CHs can better define semantic relations of POIs from low to high levels [11, 12].

Although many recent studies either exploit category transitions [9, 10] or category hierarchies [11] to achieve high-quality next POI recommendation, they neglect to consider their joint effects in a unified manner. Therefore, how to comprehensively model a user's preference transitions by exploiting CHs is a challenging problem to be solved.

Multi-task Learning based Recommendation. Multi-task learning based recommenders have shown great advantages in improving prediction accuracy and explainability, by leveraging potential correlations among multiple related tasks. In item recommendation, recent studies [21, 22] employ MTL framework to perform both rating prediction and recommendation explanation from user-generated reviews. In the next POI recommendation, several MTL neural network based methods have been proposed by integrating spatiotemporal and sequential context to perform the POI prediction task, companioned by activity prediction task [7] or time prediction task [24].

However, we argue that the potential of MTL for next POI recommendation has not been fully exploited. Inspired by the activity or location oriented sequential dependency modeling for next POI recommendation, some studies [9, 11, 32] seek to explore sequential activity patterns, so as to estimate an activity first and then predict a location based on the predicted activity. Others attempt to mine sequential location patterns via recurrent neural networks [8,33,34], due to its capability of capturing sequential information. While the presence of uncertain check-ins may weaken the sequential dependencies and result in fuzzy activities, which hinders the ability of these methods in user preference learning. Besides, existing location based MTL methods neglect the interplay between related tasks (i.e., activity and location prediction tasks) with considering the spatiotemporal context and sequential regularity. To this end, how to effectively model the context of users' check-in behaviors and characterize their underlying activities at collective POIs under the MTL framework is the core challenge to be solved.

Conversation-based Recommendation. The development of conversational recommender system (CRS) techniques [25–29] brings revolution to the existing static recommenders. The CRS typically consists of a conversation module and a recommender module, which allows the recommender to obtain a user's immediate preference and further recommends desired items via real-time conversations multiple times. Compared with static next POI recommenders, CRS enables interactions with users to acquire their immediate preference such as the desired activity and the POI type (individual or collective POI), which is one of the key point for a successful recommendation under uncertain check-in scenario. While the static recommenders under this scenario is hard to suggest accurate POIs when an undesired activity is predicted [32, 35].

Recent studies [30, 31] have demonstrated the efficacy of multi-round setting of conversations for more accurate item recommendations. However, their factorization-based recommender modules [36] adopt inner product to measure the similarity of user-item interactions, which are insufficient to capture spatiotemporal and sequential context from users' historical check-in behaviors. Hence, the key challenge is how to integrate the merit of multi-round setting of conversation module and the powerful representation ability of the recommendation module to build a location-aware CRS, so as to achieve more accurate next POI recommendations with uncertain check-ins.

1.3 Research Approaches

In Section 1.2, we have demonstrated the research challenges regarding the three recommendation problems. Next, we will illustrate the research approaches which are devised to address the issue of uncertain check-ins in the next POI recommendation.

Hierarchical Category Transition based Recommendation. Inspired by the studies which either exploit category transitions or category hierarchies to achieve high-quality next POI recommendations, we consider to model their joint effects in a unified manner, since there are different category transitions over different layers of CHs that help to infer users' preferred categories over uncertain check-ins. To this end, we propose a novel hierarchical category transition (HCT) framework based on representation learning models (e.g., Skip-gram [37]), which exploits category transitions over different layers of CHs to help thoroughly model users' preference transition. Based on the hierarchical category transition over certain check-ins, HCT is capable of predicting users' preferred categories of uncertain check-ins at collective POIs. Moreover, the hierarchical dependencies in CHs between POIs (categories) and categories could help capture better semantic relatedness and learn better representations of both POIs and categories. By sharing representations of common hierarchical categories with existing POIs, the cold start POIs could be efficiently handled.

Multi-task Learning based Recommendation. To reach the full exploitation of MTL for a more accurate next POI recommendation, we contribute a novel Interactive Multi-Task Learning (iMTL) framework by jointly modeling both types of sequential dependencies (i.e., activity and location) and the interplay between activity and location prediction prediction tasks. In particular, a two-channel encoder, i.e., temporal-aware activity encoder and spatial-aware location preference encoder, is devised to capture the latent transitions of activities and locations via long-short term memory networks

(LSTM) [38]. A fuzzy characterization strategy is proposed to better represent activity over uncertain check-ins. Then the task-specific decoder aggregates the latent representations of the two-channel encoder in an interactive manner to perform both activity and location prediction tasks. Moreover, enlightened by [39], we devise an auxiliary task of POI type (i.e., individual or collective) prediction to enhance the performance of activity prediction.

Conversation-based Recommendation. We investigate the location-aware CRS under uncertain check-in scenario, thus mitigating the inherent limitation of static next POI recommenders. We, therefore, propose a simple yet effective conversation-based adaptive relational translation framework (CART) in a multi-round paradigm[5], which consists of recommender and conversation modules. Inspired by the recent success of translation-based methods in location recommendation [40], we develop an adaptive relational translation-based recommender to model the rich context of users' historical check-ins and immediate preference in conversations. Meanwhile, the recommender adapts to user's feedback and achieves online updates by treating the rejected POIs obtained from conversations as negative samples. In this way, it can naturally alleviate the issue of the absence of the true negative samples in the static recommendation models [35, 41]. The conversation module seeks for the best conversational strategy (i.e., what attributes to ask to quickly clarify the user preference, and when to recommend POIs), aiming to achieve successful recommendations with fewer conversation turns. Moreover, it specially designs the auxiliary reward for successful collective POI recommendation and adopts the rating-based sampling strategy for individual POI selection inside such a collective POI. As such, the proposed CART naturally resolves the issue of uncertain check-ins.

Running Example. Fig. 1.3(a) depicts a running example of conversation-based next POI recommendation over uncertain check-ins. Bob is seeking a place for food to start the conversation after visiting l_2, and the system then asks the attribute of region g_1. If Bob confirms the asked region, the system continues to identify the attributes of a fine-grained category (e.g. Thai restaurant) and POI type (e.g. collective POI). Given

[5]The objective of multi-round paradigm CRS is to make a successful recommendation with less turns of conversations, where the template (e.g., Do you like __?) is used for wrapping attributes (e.g., Thai restaurant as in Fig. 1.3(a)) to simulate the conversation (e.g., Do you like Thai Restaurant?).

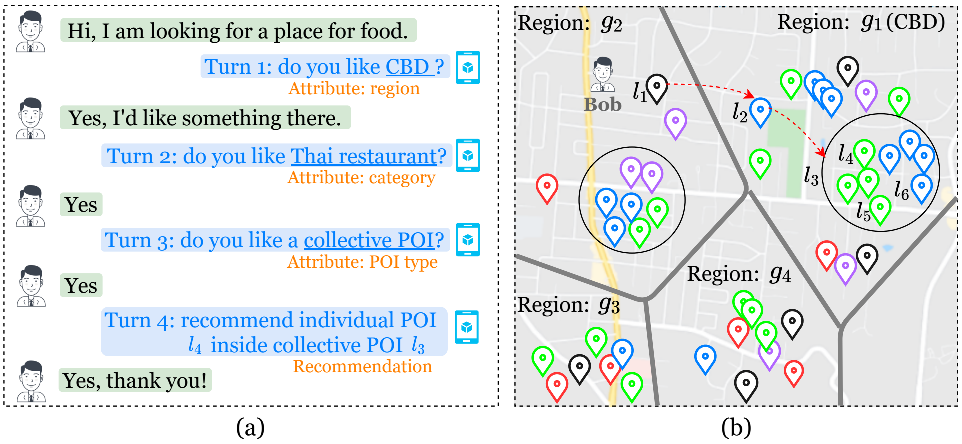


Fig. 1.3: An illustration of conversation-based next POI recommendation with uncertain check-ins. (a) An example of conversational recommendation process. (b) Examples of certain and uncertain check-ins, where the location markers are individual POIs (e.g., l_1, l_2); the POIs marked with a same color correspond to a same category, i.e., Travel (black), Shop&Service (blue), Food (green), Entertainment (purple), Nightlife spot (red); and the circle contains multiple individual POIs is collective POI (e.g., l_3).

sufficient confirmed attributes after several turns and his historical sequential check-ins, the recommender is confident to recommend a collective POI l_3 at region g_1. Meanwhile, the system further selects individual POIs (e.g., l_4) inside the confirmed l_3 via our devised rating-based sampling strategy. To this end, our CART is able to locate his preferred individual POIs via accumulating his immediate preference from a coarse- to fine-grained fashion, i.e., category-aware granularity: food → Thai restaurant; and spatial-aware granularity: region → collective POI → individual POI.

1.4 Research Contributions

The contributions of this dissertation are summarized as follows:

- We are the first to propose a new research problem, that is, predicting next individual POIs to users over uncertain check-ins, and investigate this problem by exploring and exploiting various context information from users' historical check-in behaviors to achieve more accurate recommendations.

- We first develop a novel hierarchical category transition (HCT) framework, which exploits multi-granularity category transitions derived from CHs. The hierarchical dependencies between POIs (categories) and categories are adopted to address the cold start issue, that is, recommending individual POIs inside collective POIs. We make efforts to construct three new datasets with collective POIs that are suitable for our study, based on which, we conduct extensive experiments to validate the effectiveness of our proposed HCT by comparing with state-of-the-art baselines.

- We then propose a novel Interactive Multi-Task Learning (iMTL) framework by taking advantage of two types of sequential dependencies (i.e., activity and location). We devise a two-channel encoder, i.e., temporal-aware activity encoder and spatial-aware location preference encoder, to capture the latent transitions of activities and locations. We further propose a fuzzy characterization strategy to better represent activity over uncertain check-ins. The task-specific decoder aggregates the latent representations of the two-channel encoder in an interactive manner to perform both activity and location prediction tasks. Experimental results on three real-world datasets show iMTL significantly outperforms the state of the arts w.r.t next activity and POI recommendations.

- Lastly, we investigate the task of conversation-based next POI recommendation with uncertain check-ins, highlighting the efficacy of multi-round conversations between a user and the system in predicting more accurate POIs. To our best knowledge, this is the first work to exploit CRS to address the issue of uncertain check-ins in location recommendation community, which facilitates the research of user mobility prediction. We devise the CART framework by taking advantage of the adaptive relational translation-based method and CRS. That is, CART consists of the tailored recommender module and the conversation module, seeking to model a user's rich context of historical check-ins and immediate preference in conversations to achieve successful next POI recommendation with fewer conversation turns. We conduct extensive experiments to evaluate the performance of our proposed CART framework on four real-world datasets. The results demonstrate the superiority of CART over both state-of-the-art CRSs and static recommendation methods for next POI recommendation over uncertain check-ins.

Overall, to achieve the next POI recommendation with uncertain check-ins, we propose a series of novel recommendation approaches by exploiting various context information from users' historical check-in behaviors. First, by exploiting both effects of the category transition and category hierarchy, a representation learning based approach HCT is proposed; Second, although the pioneer effort HCT is able to address the issue of uncertain check-ins to some extent, it fails to well characterize the underlying activities over uncertain check-ins and model the interplay between sequential activities and locations, thus an interactive multi-task learning framework iMTL is devised; Third, to address the intrinsic limitation of existing static next POI recommenders, we proposed a conversation-based recommendation method CART to help clarify users' immediate preferences and make next POI recommendations with fewer conversation turns.

1.5 Dissertation Organization

The rest of the dissertation is organized as follows. In Chapter 2, we review the state-of-the-art studies related to our research work. To achieve the next POI recommendation with uncertain check-ins, Chapter 3 first presents HCT approach by jointly modeling both category transitions and category hierarchies. Next, Chapter 4 introduces the iMTL model to learn the two types of sequential dependencies by exploring users' underlying activity over uncertain check-ins. Chapter 5 illustrates the proposed CART for conversation-based next location recommendation. Finally, in Chapter 6, we conclude this dissertation and point out several directions of future work.

Chapter 2

Literature Review

As introduced in Chapter 1, this dissertation aims to achieve the next POI recommendation with uncertain check-ins by exploiting various context features from users' historical check-in behaviors. We first provide an overview of state-of-the-art studies that incorporate the context features for next POI recommendation. Then, we focus on research efforts related to the three recommendation problems that have been defined in Chapter 1, i.e., hierarchical category transition based recommendation, multi-task learning based recommendation and conversation-based recommendation, and point out their advantages and limitations in resolving the issue of uncertain check-ins.

2.1 Context Features for Next POI Recommendation

With the development of the location-based social networks (LBSNs), users would like to share their daily behaviors by checking in POIs, thus delivering a large amount of user mobility records and promoting the location recommendation tasks on LBSNs, such as general POI recommendation and next POI recommendation. Different from the traditional item recommendation tasks (e.g., movies and goods), POI recommendation is highly context-dependent, e.g., spatial and temporal contextual information is the basis for representing user mobility [5, 42–46]. In this section, we review the research studies that incorporate different context features on the POI recommendation task, especially the next POI recommendation task. We categorize them based on the context they exploit, i.e., categorical context, spatial context, temporal context and sequential context.

2.1.1 Categorical Context

POIs are often characterized by a hierarchical category tree, as illustrated in Fig. 1.2. Obviously, category information reflects the activity/intention (e.g., food and entertainment) of a user's check-in behaviour at a specific location, which plays an important role in alleviating the data sparsity issue [41]. To this end, many next POI recommendation methods model the user preference over the categories of POIs to promote the recommendation accuracy. For instance, a matrix factorization based method [11] aims to factorize a user-category transition matrix to predict the next category first as the user's future activity, and then predict POIs corresponding to such a predicted category. Then, a two-fold approach Listwise Bayesian Personalized Ranking (LBPR) [9] models category lists by employing a third-rank tensor, so as to predict user's category-level preference first, then derive POIs by considering both spatial context and categorical ranking influence. Inspired by the RNN based methods in modeling sequential location-level behaviors, a semantic-enriched recurrent model (SERM) [6] models user intentions by fusing activity semantics, spatiotemporal context and user preference in a unified way. Considering the interplay between activity and location preferences, a multi-task framework MCARNN [7] leverages the spatial-activity topic to perform both activity and location predictions. More recently, a personalized long- and short-term preference learning (PLSPL) is proposed to learn user's location-level and category-level preference via two parallel LSTM units.

2.1.2 Spatial Context

In LBSNs, spatial context (i.e., geographical influence), derived from users' historical check-in records has shown effectiveness in improving location recommendation performance. According to Tobler's First Law of Geography "Everything is related to everything else, but near things are more related than distant things" [47], a user intuitively prefers to visit nearby POIs than distant ones and tends to explore nearby POIs of a POI that he is in favor of. To model the spatial context, various methods have been developed.

They can be broadly categorized into distance-based methods and region-based methods. For distance-based methods, early studies [1, 48, 49] employ power-law distribution to model the check-in probability regarding the distance between two locations visited by a user. Due to the limitation of the above prior distribution in modeling the personalized spatial preference on POIs, some efforts, such as Rank-GeoFM [50], PRME [4] and LBPR [9], use the distance-based weight factor to fuse the geographical influence and thus generate a ranking list of POIs. Other studies, such as ATST-GRU [51] and ATST-LSTM [34], learn a low-dimensional dense vector to represent the geographical distance. Region-based methods assume that POIs in the same region share a similar attraction to users. Generally, they divide the whole geographical space into various geographical regions and characterize the region-level influence [2, 52–54], so as to enhance the POI recommendation and next POI recommendation tasks.

2.1.3 Temporal Context

Users exhibit different check-in preferences at different hours of a day, e.g., one may often visit restaurants during lunchtime, while the other prefers the nightlife spot at midnight. This motivates us to investigate such a significant property of user mobility over time. Many studies [33–35, 49, 55, 56] model this property by representing the exact time (i.e., dividing a day into 24 hours/time slots) or time interval between two location visits. For example, the early time-aware POI recommendation [49] splits time into hourly-based slots and models the temporal preference over POIs, thus suggesting the POIs for a user at a specific time. Analogously, the time-enhanced matrix factorization method LRT [55] models user preference at different time by utilizing temporal regularization and temporal aggregation. The recent RNN-based method PLSPL [56] tends to represent each time slot with latent vector to help capture the temporal influence on categories and locations, and the extended RNN framework ST-RNN [33] seeks to model local temporal context with time-specific transition matrices for different time intervals. More importantly, most next POI recommendation approaches typically incorporate both spatial and temporal contexts to help learn better user preference and thus achieve more accurate next POI recommendations.

2.1.4 Sequential Context

Sequential context (i.e., sequential regularity) has been widely investigated in the next POI recommendation since user mobility exhibits sequential transitions, e.g., users usually go to shops after checking in at food places. Existing sequential POI recommenders typically model users' sequential regularities from historical check-in trajectories by employing first-order (i.e., Markov chain property) or high-order properties. Based on the Markov chain property, the early work FPMC-LR [3] models two successive check-ins with localized region constraint incorporated. The studies derived from metric embedding, e.g., PRME [4], and Word2Vec, e.g., CAPE [57], attempt to model the sequential check-ins and learn transition regularities by capturing the local co-occurrence of POIs within a check-in trajectory. With the success of the RNN in capturing high-order sequential information, RNN and its variants have been adopted to model the whole sequential behaviors for next POI recommendation [6, 33, 34, 58–60]. For instance, ST-RNN [33] employs the recurrent structure to model the trajectory by considering time-specific and distance-specific transition matrices to characterize spatiotemporal context. To capture the spatiotemporal relations of neighbor check-ins and user's long- and short-term preference, STGN [8] is devised to learn user's sequential check-in behaviors under LSTM architecture in a gate mechanism way. Later, ARNN [59] extends the LSTM with an attention mechanism to capture the relevant behaviors within trajectories.

Existing next POI recommendation approaches typically model the joint effects of multiple context features to learn user preference from the historical check-in trajectories. Therefore, in this dissertation, we focus on exploiting the above context features to address the issue of uncertain check-ins in the next POI recommendation. In the following, we will provide a detailed literature review regarding the three recommendation problems defined in Chapter 1.

2.2 Hierarchical Category Transition based Recommendation

This section provides an overview of state-of-the-art next POI recommenders that are highly related to our hierarchical category transition based recommendation. They can be broadly classified into two categories, namely latent factor model based approaches and representation learning model based approaches.

2.2.1 Latent Factor Model based Approaches

Some studies focus on designing approaches based on latent factor models (LFMs) [61] by employing sequential information between consecutive check-ins to achieve better next POI recommendations. Specifically, based on Markov chain property [62], Cheng et al. [3] exploit the tensor factorization method to model the user's personalized check-in transition. Zhang et al. [63] predict the probability of next location by additive Markov chain. Zhao et al. [5] propose a ranking-based pairwise tensor factorization framework to model successive POIs. However, these methods directly model users' check-in sequences, thus failing to mine users' insights of check-in locations as well as the preference transition patterns.

To this end, other studies endeavor to exploit the transition pattern of categories of POIs to predict next POI. For instance, Liu et al. [11] investigate the patterns of users' preference transitions over location categories, whereby a two-fold approach based on matrix factorization is proposed. In particular, they first predict next preferred categories for each user, based on which they further estimated precise POIs in the corresponding categories. Later, He et al. [9] also propose a two-fold approach to predict the preferred next category by using the tensor factorization method, and then predicted the next POI based on the predicted category preference. Empirical studies have proven that users' category transitions are more effective in reflecting the insights of users' transition patterns, thus achieving high-quality next POI recommendation.

We, therefore, advance the state of the arts by developing HCT, which simultaneously leverages category transition at different layers of the category hierarchy as well as the hierarchical dependencies between POIs (categories) and categories, to help resolve our proposed new research problem, that is, recommending next precise individual POIs with uncertain check-ins. By doing so, the proposed HCT can not only capture users' preference transition in different granularity, but also better address the cold start issue.

2.2.2 Representation Learning Model based Approaches

Representation learning models (RLMs) have attracted a considerable amount of interest from various domains, with recommender systems being no exception [64, 65]. The popularization of RLMs in recommendation can be mainly attributed to the Word2Vec techniques (e.g. CBOW and Skip-gram [37, 66]) originated from the natural language processing (NLP) domain. Word2Vec has shown to be effective in capturing semantic representations of words by modeling the context correlations of words in sentences. RLMs have also been widely extended in next POI recommendation [53, 57, 67–69], and have proven to be more effective than LFM based approaches [5, 68].

The pioneer work in [67] directly employs the Word2Vec to learn the representations of POIs by considering POIs as words and all check-ins of a user as one sentence, for next POI recommendation. Later, Liu et al. [68] first divide the whole check-in sequence of a user into several sequences by different time frames (i.e., hour, day and week). Then Skip-gram is applied to learn representations of POIs. Besides, temporal influence is incorporated in their proposed method. Similarly, Zhao et al. [69] propose a temporal POI embedding model that considers users' daily check-in sequences. After that, Feng et al. [53] present a POI2Vec model which learns the POI sequential transition of users. Furthermore, they develop a binary tree over the POIs to incorporate geographical influence. Recently, Chang et al. [57] propose a content-aware POI embedding model which exploits the check-in context (i.e., successive check-ins) and text content (i.e., descriptions of POIs), to capture both the sequential transition pattern and characteristics of POIs.

Despite the satisfactory performance achieved by the above methods, they cannot better capture and model the insights of users' preference transition patterns. Meanwhile, they are not able to better handle the extremely cold start issue in next POI recommendation. In contrast, we propose a hierarchical representation learning framework - HCT, which aims to learn better representations of POIs by simultaneously considering category transitions at different layers of the category hierarchy, as well as the hierarchical dependencies of POIs (categories) and categories. In doing this, the proposed HCT is capable of better modeling users' preference transition patterns over the uncertain check-ins and handling the cold start issue caused by the individual POIs inside collective POIs.

2.3 Multi-task Learning based Recommendation

Multi-task learning tends to enhance learning efficiency and prediction accuracy by learning multiple related objectives. Location-aware recommendations usually refer to multiple related prediction tasks, such as activity and POI predictions. In this section, we present the related studies with respect to activity and location aware recommendation as well as multi-task learning for recommendation.

2.3.1 Activity and Location aware Recommendation

In contrast to the next POI recommenders that solely model user's location-level preference, many methods that consider both effects of category-level and location-level preference have shown to be effective in capturing more accurate user preference. They can be generally classified into two categories: Markov-based models (MMs) and deep learning based models (DLMs).

Markov-based based Models. Markov-based models (MMs) aim to learn the transition probability of successive movements. A line of research models the category transition patterns and performs next POI recommendations in a two-fold manner, i.e., predict the most likely locations based on the inferred categories. An obvious advantage of this two-fold approach is to reduce prediction space of locations. Liu et al. [11] introduce a category-aware POI recommender, which combines MMs with matrix factorization to learn the transition regularity by factorizing user-category and user-location matrices. It predicts possible categories of next check-in locations and possible POIs corresponding to the predicted categories. Meanwhile, it groups similar users by considering temporal and category effects to improve the accuracy of category-level preference transition prediction. Analogously, Ye et al. [70] propose a mixed hidden Markov model to model the transition pattern and dependency between a user's check-ins at the category level, so as to predict the category for the next step and then suggest a location given the category distribution. Later, He et al. [9] develop a Listwise Bayesian Personalized Ranking (LBPR) approach to model the probability of category transition based on the first-order Markov property. The proposed LBPR predict POIs by incorporating the spatial influence and category ranking influence according to the next category prediction results.

However, these methods fail to incorporate the temporal regularity to infer user's activity preference. To address this limitation, Liao et al. [71] propose a Context-Aware Hybrid (CHA) approach, which employs kernel density estimation to model the time variation of location preference and designs a two-stage method to predict the next location based on the inferred activity.

Deep Learning based Models. Deep neural networks have shown great potential in various domains, e.g., speech recognition, natural language processing and recommendation system. Recurrent neural networks (RNNs) and their variants are especially widely explored in the next POI recommendation. Several RNN-based methods which consider user's activity preference have been proposed to date. For instance, Yao et al. [6] propose a semantics-enriched recurrent model (SERM), which jointly models spatiotemporal regularities and activity semantics by using a recurrent neural network. It is capable of capturing activity preference over the check-in locations and achieving more accurate next POI recommendation. Recently, Wu et al. [56] develop a long- and short-term preference learning method, which models the contextual features of POIs in users' check-in records by leveraging an attention mechanism, and learns the different influences of locations and categories by training two parallel LSTM models. Yu et al. [41] devise a category-aware deep model by incorporating categorical and geographical influence to reduce search space via a two-layer filter architecture.

Despite their success for the next POI recommendation, all the above methods fail to mine the transition patterns due to the presence of uncertain check-ins and suffer from the limitation of: (1) well characterizing the underlying activity over uncertain check-ins; and (2) exploiting the spatial and temporal contexts in a fine-grained way.

2.3.2 Multi-Task Learning for Recommendation

A line of research focuses on generating explanations about the recommended items by designing multi-task learning methods. For example, Wang et al. [72] develop a multi-task learning method for the explainable recommendation, which combines two tasks of item recommendation and textual explanation generation by adopting a joint tensor factorization. In order to improve the interpretability of latent factor models, Lu et al. [21] propose to perform rating prediction and recommendation explanation by integrating matrix factorization and adversarial sequence to sequence learning model. It is capable of generating personalized reviews for each user and item, thus helping to reveal personalized user preferences and item attributes. To better model the correlation between the rating prediction and recommendation explanation, Chen et al [22] propose a co-attentive multi-task learning (CAML) model by designing an encoder-selector-decoder architecture, where the hierarchical co-attentive selector could effectively select cross knowledge that is important for user-item interactions.

In the next POI recommendation, there are also some related studies that consider different task combinations to enhance the performance POI prediction. For instance, early work [7] designs a multi-task context aware RNN based model (MCARNN) by considering users' activity and location preferences simultaneously for both activity and location recommendation. It adopts a gated RNN as the shared hidden layer for modeling sequential dependency and temporal regularity from the sequential check-in data. Different from the MCARNN that only utilizes the historical temporal information, Zhong et al. [24] attempt to investigate the prediction of the user's next check-in time to improve POI recommendation, whereby they develop a multi-task neural network recommendation model (MTNR) to jointly learn when and where users prefer, i.e., learning user's next POI preference from the next check-in time prediction. To better capture the semantic motivation of users' check-in behaviors, Feng et al [73] propose a multi-task attentional recurrent neural network model, which performs next POI recommendation cooperating with next activity type prediction and next check-in time prediction.

Despite their success for next POI recommendation, all existing methods fail to explore either the underlying activities over uncertain check-ins or interplay between correlated tasks. In contrast, our proposed multi-task learning framework iMTL is designed to learn temporal-aware activity preference and spatial-aware location preference for activity and location predictions. It first represents users' uncertain activities via a fuzzy characterization strategy, then models the interplay between activity and location for more accurate next POI recommendation, meanwhile, an auxiliary task of POI type prediction is devised to enhance the performance of activity prediction.

2.4 Conversation based Recommendation

Most existing next POI recommenders (i.e., static methods) focus on learning user's accurate preference for his next movement by exploiting various context features. However, such static methods are hard to capture user's immediate preference and incorporate user's feedback regarding the recommendation results. Conversation based recommendation bridges the gap between the static methods and user's feedback. This section first reviews the state-of-the-art static next POI recommendation methods that exploit the different context effects, then focuses on conversational recommender systems.

2.4.1 Static Next POI Recommendation

Existing static next POI recommendation task is formulated as estimating a set of user-POI affinity scores [7, 34, 46, 74] without interacting with users. It is typically achieved by mining rich context information from users' historical trajectories, such as spatiotemporal context [42, 44, 54], sequential regularity [8, 58, 75], categorical influence [9, 41] or their joint effects [56,59,60,73], without interacting with users. By doing so, these static methods are able to learn user preference on future movements. Despite their great success for inferring user's future preference on next location visit by modeling rich context effects, these efforts posit that users' accurate check-ins at individual POIs (i.e. certain check-ins) are always available and accessible, ignoring the presence of uncertain check-ins, which brings the issue of incomplete context information of check-ins and thus limits the capability of the recommender system in decision-making [76]. To ease such an issue, recent studies turn to model the context-aware sequential regularity as well as to characterize the underlying activity over uncertain check-ins [32, 35]. However, these static recommendation approaches face an intrinsic limitation: they cannot well capture user's immediate preference solely based on his historical behaviors, since they are not able to interactively clarify a user's current preference in the static model setting. It is, therefore, hard to locate the desired POI when the system fails to predict accurate user's activity (e.g., category) for the approaches in [9, 32, 35, 41]. As collecting feedback on whether the recommended POI satisfies the user demand could help enhance the recommender systems [77], this directly motivates us to exploit a conversational technique in the next

POI recommendation over the scenario of uncertain check-ins, due to its promising ability in capturing immediate user preference through an interactive conversion.

2.4.2 Conversational Recommender System

The conversational recommender system (CRS) has been widely explored under different assumptions and application scenarios [78], which provides a new possibility for capturing immediate user preference via dynamic interaction with users [25, 26, 28, 29, 79–89]. For example, a line of research [29, 79, 90] adopts the bandit methods (e.g., Thompson Sampling [91]) to solve the exploit-explore issue in a CRS for cold start users. Other works that incorporate natural language processing module [82, 83, 92–94] aim to understand the user preference from their utterances and generate fluent responses for natural and effective dialogues. As the CRS needs to interact with users multiple times for asking the user preference on attributes or making recommendations, a good CRS should learn a conversational strategy for when to ask or make recommendations.

There is another line of research that is closely related to our study. In particular, Sun et al. [25] propose a conversational recommender model (CRM) to interactively ask user clarification questions and make personalized recommendations. The CRM trains a deep policy network to decide whether to ask an attribute or make recommendations at each turn under the single-round setting (i.e., the CRS terminates once making a recommendation regardless of the result is satisfactory or not). Moreover, the situation for interactions between the user and system is complex in real applications. For example, considering the user's patience, the CRS should strategically ask questions to quickly clarify the user's current preference and achieve successful recommendations with fewer conversation turns. To this end, recent studies focus on learning a conversational strategy and taking advantage of a multi-round setting for better recommendation, i.e., the CRS interacts with a user and makes recommendation multiple times until the user accepts the recommendation.

The early study [30] propose a new CRS framework, i.e., Estimation-Action-Reflection (EAR), which mainly handles the interactions between the conversational component and recommender component under the multi-round setting and solves three fundamental problems as follows: 1) What attributes to ask? 2) When to recommend items? and

3) How to adapt to users' feedback? Later, the same authors develop a conversational path reasoning framework by modeling the conversational recommendation as an interactive path reasoning problem on a user-item-attribute graph [31], with the assumption that exploiting the attribute preference in an explicit way can better carry forward the advantages of CRS. Inspired by the success of CRS with the multi-round setting, we formalize our problem under this research line, whereby the CRS focuses on simulating such a scenario instead of real dialogue: the system interacts with a user via asking attributes (e.g., Thai restaurant) wrapped by the template (e.g., Do you like __?) multiple times to clarify his preference and make recommendations until the user accepts the recommendation results or chooses to quit [30].

We argue that an efficient location-aware CRS should delicately accommodate both rich context of users' check-in behaviors and the merit of multi-round CRS, so as to ease the issue of uncertain check-ins in next POI recommendation. As such, our proposed CART advances the state-of-the-art CRSs in two aspects: (1) the recommender module models a user's historical check-ins and immediate preference obtained in conversations, which helps safely locate the accurate preference for a user's next movement; and (2) the conversation module aims to learn the conversational strategy and achieve successful recommendations with fewer turns. Most importantly, it specially designs the auxiliary reward for successful collective POI recommendations and adopts the rating-based sampling strategy for individual POI selection inside such a collective POI. In sum, this results in a new angle of CRS for the next POI recommendation over uncertain check-ins.

2.5 Summary

This chapter presents an overview of the state-of-the-art efforts related to our study. We first briefly introduce the context features that are widely explored in the next POI recommendation. Then, we review the existing studies corresponding to the three recommendation problems defined in Chapter 1, i.e., hierarchical category transition based recommendation, multi-task learning based recommendation and conversation based recommendation. To address the limitations of existing studies, we propose a series of recommendation approaches to resolve the problem of the next POI recommendation with uncertain check-ins.

(1) Most existing next POI recommenders ignore to consider both effects of category transition and category hierarchy. To help infer user preference over uncertain check-ins and alleviate the cold start issue, We, therefore, develop a framework HCT to exploit category transitions over different layers of category hierarchies and hierarchical dependencies in Chapter 3.

(2) Most existing multi-task learning neural network based recommenders are not sufficient to model users' sequential dependencies of the activity and location, as they neglect the interplay between activity and location for the next POI recommendation. To this end, we devise an iMTL to better learn users' location preferences and activity preference in an interactive manner in Chapter 4.

(3) Most conversation-based recommenders are not sufficient to model the rich context information of users' check-in behaviors, as they generally adopt the inner product to measure the similarity of user-item interactions. Therefore, in Chapter 5, we propose a location-aware conversational recommender CART model users' context of location visits and immediate preference in conversations to achieve successful next POI recommendation with fewer conversation turns.

Chapter 3

HCT: Hierarchical Category Transition Method[1]

User's preference transition over the categories of POIs (i.e., category transition) has been exploited to alleviate the sparsity issue in check-in data, as it is capable of deeply reflecting the semantic of user's transition pattern. By studying the patterns of user preference transition over categories of certain check-ins, we are able to infer possible categories of uncertain check-ins. Moreover, POIs are typically organized by the category hierarchies (CHs), where the dependencies between POIs (categories) and categories at different layers of CHs can help define their semantic relations. Modeling the CHs has been illustrated to be effective in easing the cold start issue. Although most existing studies either exploit the category transition or the category hierarchy to improve the effectiveness of the next POI recommendation, they are insufficient to capture user preference due to the limited ability for modeling their joint effects in a unified manner.

In this chapter, we propose a novel hierarchical category transition (HCT) framework which exploits multi-granularity category transitions derived from category hierarchies to capture more accurate user preference. Then the hierarchical dependencies in CHs assist in learning better representations of both POIs and categories. By sharing representations of common hierarchical categories with existing POIs, the cold start POIs could be efficiently handled. The experimental results verify the superior performance of our proposed HCT against state-of-the-art approaches.

[1]The work in this chapter has been accepted by Information Sciences [32], 2020.

The rest of this chapter is organized as follows. In Section 3.1, we first provide definitions of concepts, data construction and analysis. We then formulate the investigated research problem in Section 3.2 and demonstrate the proposed HCT framework in Section 3.3. Section 3.4 presents the experimental results, followed by the conclusion in Section 3.5.

3.1 Data Construction and Analysis

3.1.1 Data Construction

For ease of presentation, we first provide definitions of concepts utilized in this work, as follows:

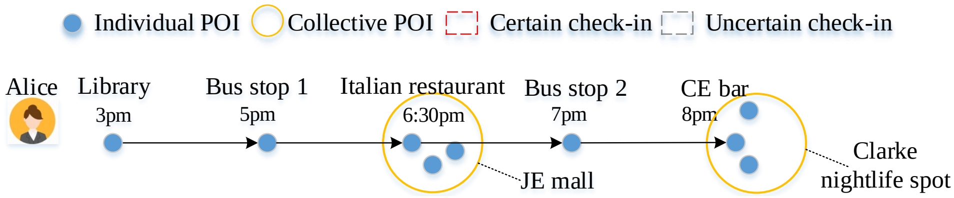
Definition 1 ***(Collective POI)*** *It denotes the gathering place with a number of individual POIs. Typical examples include shopping mall, plaza, university, beach and garden center, nightlife spot, etc. For instance, a shopping mall is one kind of collective POIs, which contains various restaurants, coffee bars, clothes shops, and supermarkets, etc.*

Different from existing next POI recommendation studies, each individual POI inside a collective POI is unobserved in the users' check-in sequences during the training procedure of our study. In other words, we can only observe a collective POI instead of a precise POI in a user's check-in trajectory, if the visited POI is inside a collective POI.

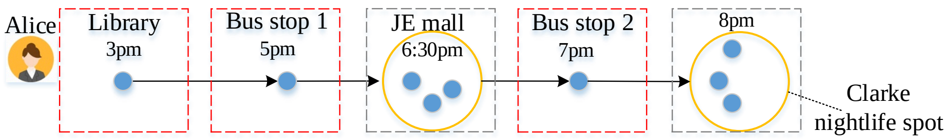
Definition 2 ***(Individual POI)*** *It denotes the single POI, such as restaurants, coffee bars, and clothes shops, located either inside or outside collective POIs.*

Definition 3 ***(Certain Check-in)*** *If a user visits an individual POI outside the collective POI, we name it as a certain check-in.*

Definition 4 ***(Uncertain Check-in)*** *If a user visits an individual POI inside a collective POI, we name it as an uncertain check-in, since we cannot access the precise individual POI visited by the user inside the collective POI.*



(a) An example of check-in data used in existing models



(b) An example of check-in data that is available in reality.

Fig. 3.1: A running example to illustrate Alice's check-in sequences: (a) shows the sequence that is utilized in most existing next POI recommendation approaches: Library → Bus stop 1 → Italian restaurant → Bus stop 2 → CE bar; and (b) depicts the sequence that may be available in reality: Library → Bus stop 1 → JE mall → Bus stop 2 → Clarke nightlife spot.

Let us consider the running example in Fig. 3.1 to further explain these concepts. The check-in sequence of Alice is: Library → Bus stop 1 → JE mall → Bus stop 2 → Clarke nightlife spot. In this case, JE mall and Clarke nightlife spot are *collective POIs*; Library, Bus stop 1, Bus stop 2 and all the single POIs inside JE mall and Clarke nightlife spot (e.g., Italian restaurant and CE bar) are *individual POIs*. Alice's check-ins on Library, Bus stop 1 and Bus stop 2 are *certain check-ins*, whereas her check-ins on JE mall and Clarke nightlife spot are defined as *uncertain check-ins*.

There is, however, no available public dataset containing collective POIs that is suitable for our research problem. We, therefore, construct new datasets based on the widely utilized Foursquare dataset [95], via grouping individual POIs in different semantic regions into collective POIs[2]. Foursquare is one of the most popular location-based social networks (LBSNs), which records 33,278,683 check-ins by 266,909 users on 3,680,126 POIs in 415 cities of 77 countries from April 2012 to September 2013. As users' check-ins are always centered around a certain region (e.g., a city), we follow the state of the

[2]Note that we assume all accurate check-ins inside collective POIs are missing and replaced by uncertain check-ins. Although users may randomly leave some accurate check-ins inside collective POIs, it's hard to construct a dataset that satisfies such more real user behaviors. Therefore, our study explores the low-bound performance for the next POI recommendation under this assumption.

Table 3.1: The statistics of three original POI datasets, where 'SIN, NYC, LA' refer to Singapore, New York City, and Los Angeles, respectively.

	# User	# POI	# Check-in	# 1st-Layer Category	# 2nd-Layer Category
SIN	2676	3440	116757	264	177
NYC	1982	4243	187750	210	165
LA	2109	2195	70189	167	103

arts [9, 53], and collect three POI datasets from Singapore, New York City and Los Angeles. Additionally, we pre-process these datasets by filtering out users with fewer than 10 check-ins, as well as POIs with fewer than 10 visitors as in [4, 57].

Besides, two-layer category hierarchies (CHs) of POIs are obtained through the Venue API[3] of Foursquare. Note that the originally obtained CH of Foursquare is unbalanced, where different POIs may belong to hierarchical categories with different depths. For instance, from bottom to top, the POI 'Baiano Bar Restaurant' is labelled with a 5-layer hierarchical category: Baiano Restaurant → Brazilian Restaurant → South American Restaurant → Latin American Restaurant → Food, where 'Baiano Restaurant' is the first-layer (more localized) category and 'Food' is the fifth-layer (more general) category; while 'Chinese restaurant → Asia restaurant' is the two-layer CH of 'Din Tai Fung' (POI), where 'Chinese restaurant' is the category at the first layer, and 'Asian restaurant' is the category at the second layer. For simplicity, following [11, 14], we trim the CH by only keeping the bottom-two category layers for each POI. This makes sense from two aspects: (1) the categories at higher layer always represent larger concepts, which may inevitably introduce noise leading to inaccurate category transition patterns [12]; and (2) based on the preliminary data analysis, we find that the minimum layer (depth) of CH for all the POIs is two. That is to say, all the POIs have a two-layer CH at least, while there are only 72% of POIs with three-layer CH, 14% of POIs with four-layer CH, and 2% of POIs with five-layer CH. Therefore, we adopt the two-layer balanced CH to cover all the POIs in our study.

The statistics of three original datasets are summarized in Table 3.1. Based on the three original datasets, we construct new datasets with collective POIs via Google Map in four steps. First, we find out the geographical center of each collective POI (e.g., shopping mall) as the seed. In particular, we consider collective POIs with different

[3] `https://developer.foursquare.com/docs/resources/categories`

Table 3.2: A running example of mapping the original check-ins to newly-constructed check-ins for Alice in SIN.

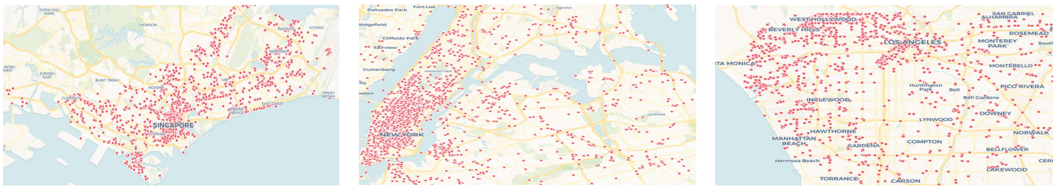
	1st	2nd	3rd	4th	5th
Original	Library (individual)	Bus stop 1 (individual)	Italian restaurant (individual)	Bus stop 2 (individual)	CE bar (individual)
New	Library (individual)	Bus stop 1 (individual)	JE mall (collective)	Bus stop 2 (individual)	Clarke nightlife spot (collective)

Table 3.3: The statistics of three newly-constructed POI datasets, where 'SIN, NYC, LA' refer to Singapore, New York City, and Los Angeles, respectively.

	# User	# POI	# Check-in	# 1st-Layer Category	# 2nd-Layer Category
SIN	2676	1633	116757	234	177
NYC	1982	2454	187750	179	152
LA	2109	1576	70189	150	103

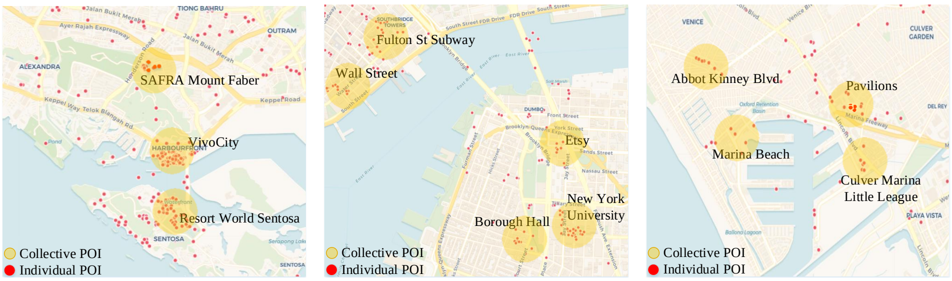
semantics, including shopping mall, plaza, university, beach, garden center, nightlife spot, amusement park, etc. Second, we group individual POIs which gather around the seed within a distance threshold in a coarse- to fine-grained way. We empirically set different thresholds, i.e., $0.5km$, $0.8km$ and $1km$ for Singapore, New York City and Los Angeles, respectively, to cover as many individual POIs as possible (coarse-grained way). Third, to ensure the quality of constructed collective POIs, we manually check whether the individual POIs grouped by the second step are correctly classified into the right collective POIs (fine-grained way). Lastly, for all user-POI check-in records, we replace all individual POIs by their associated collective POIs, if the individual POIs are inside collective POIs. By doing so, the total records of user-POI check-ins in the new datasets retained the same as that of the original datasets. Table 3.2 depicts an example of mapping the original check-ins to the new ones.

Moreover, we also label all newly-constructed collective POIs with hierarchical categories. The 1st-layer categories are just their corresponding semantics, such as shopping mall, plaza, nightlife spot, university, etc. The 2nd-layer categories are the existing ones in Foursquare, such as shop & service, entertainment, etc. For instance, the two-layer CHs for JE Mall and Clark night spot in the running example from bottom to root are: JE Mall → shopping mall → shop & service; and Clark nightlife spot → nightlife spot → entertainment. The statistics of three newly-constructed datasets are summarized in Table 3.3.



(a) SIN (b) NYC (c) LA

Fig. 3.2: All individual POIs marked by red dots for the three POI datasets.



(a) SIN (b) NYC (c) LA

Fig. 3.3: Examples of collective POIs and individual POIs for the three POI datasets.

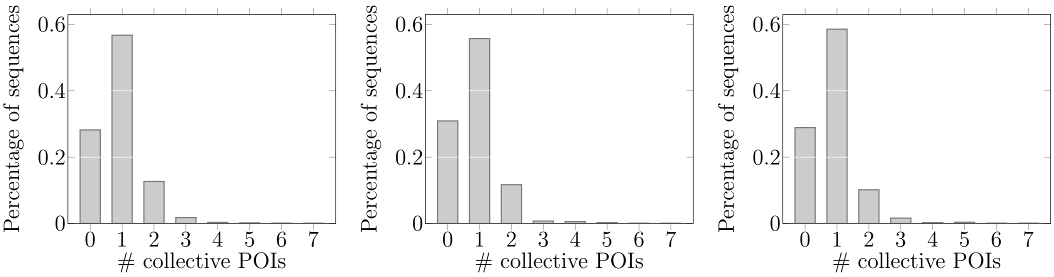
It is worth mentioning that our defined collective POI is different from the geographical region proposed by [52], in two aspects: (1) one geographical region is always associated with a specific function, such as education or business. In contrast, each collective POI may contain various POIs with different functions. For example, a shopping mall consists of restaurant, clothes shops, supermarkets and book store, etc; and (2) the users' precise check-ins in geographical region can be completely observed in [52]; whereas we cannot access the precise individual POIs that a user visited in the collective POI.

3.1.2 Empirical Analysis

We conduct extensive analysis on the three real-world POI datasets to obtain insightful guidance that assists in resolving the proposed research problem.

Presence of Collective POIs. To validate the presence of collective POIs in reality, we locate and visualize all the POIs in the three original datasets by Google Map[4]. Fig. 3.2 shows the distributions of original individual POIs in SIN, NYC, and LA, respectively. Several interesting findings are obtained: (1) we observe that some POIs are indeed

[4] `https://www.google.com/maps`



(a) SIN (b) NYC (c) LA

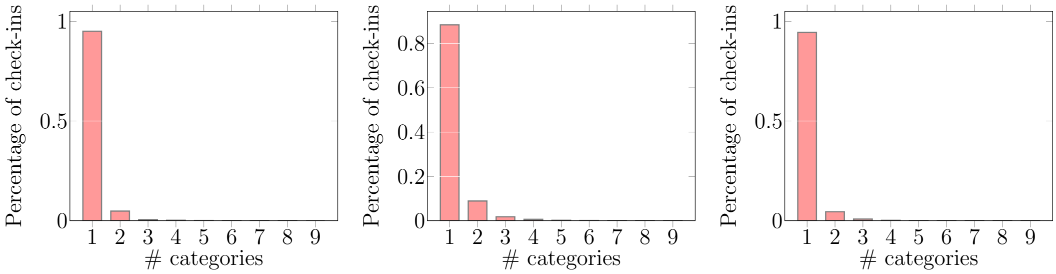
Fig. 3.4: Distribution of sequences over the number of collective POIs visited by day.

densely located in a certain area, while some are not; (2) we find that most of the regions with dense POIs located in are collective POIs, such as shopping mall, university or beach. Fig. 3.3 shows examples of individual POIs (marked by red dots) and collective POIs (marked by yellow dots) in SIN, NYC and LA, where we can observe different kinds of collective POIs, for example, VivoCity mall in Fig. 3.3(a), New York University in Fig. 3.3(b) and Marina Beach in Fig. 3.3(c); and (3) through calculation, we further notice that there are around 60%, 50%, 30% individual POIs which are inside collective POIs for the three datasets, respectively. These findings help verify the presence of collective POIs and the significance of our proposed research problem.

Uncertain Check-in Patterns on Collective POIs. In the new datasets, for all user-POI check-in pairs, the visited individual POI is replaced by its corresponding collective POI, if it is inside the collective POI, as shown in Table 3.2. Therefore, the total number of user-POI check-in pairs keeps the same as the original datasets. Based on the three new datasets, we analyze users' check-in behaviours over collective POIs in a day.

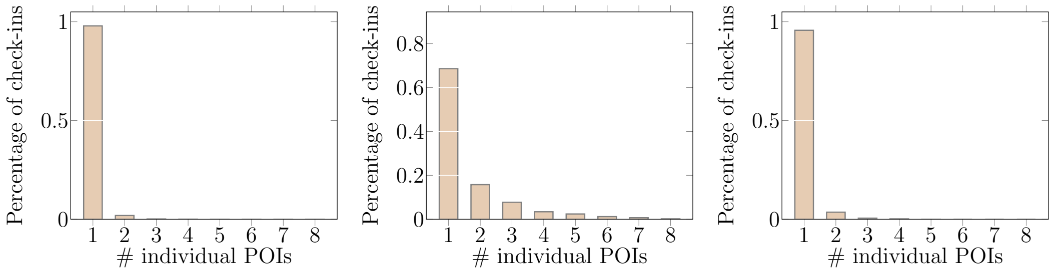
We first study users' daily check-ins over the number of collective POIs. In particular, we split the check-ins of each user into sequences by day, and then count the number of unique collective POIs that appear in each sequence. Fig. 3.4 shows the percentage of sequences w.r.t. the number of collective POIs. We observe that around 70% check-in sequences include at least one collective POI, and most of them include less than three collective POIs, on the three datasets.

We then investigate the number of unique categories of individual POIs visited by users within a collective POI. For the consecutive check-ins at the same collective POI



(a) SIN (b) NYC (c) LA

Fig. 3.5: Distribution of check-ins over the number of visited categories within a collective POI.



(a) SIN (b) NYC (c) LA

Fig. 3.6: Distribution of check-ins over the number of individual POIs visited under each category within a collective POI.

of each user, we count the number of categories that the visited individual POIs belong to. The results are shown by Fig. 3.5. We notice that more than 95% of check-ins at collective POIs are associated with no more than two categories, on the three datasets. Finally, we study the check-in distribution over the number of individual POIs visited under each category within one collective POI by users. The results are depicted by Fig. 3.6, where we find that more than 80% of check-ins at collective POIs have no more than 2 visited individual POIs under each category, on the three datasets. These findings provide invaluable guidance for selecting collective POIs, categories as well as individual POIs inside the selected collective POIs when generating recommendations, which will be introduced in Section 3.3.

Table 3.4: Notations for HCT.

Notations	Descriptions
u	User u
l_i, l_k, l_g, l_m, l_n	POIs l_i, l_k, l_g, l_m, l_n
$\mathcal{H}$	Category hierarchy
$N_{\mathcal{H}}$	Number of layers of the category hierarchy $\mathcal{H}$
$c_{l_i}^j$	Category of POI l_i at layer j
$\mathcal{D}_l$	POI set
$\mathcal{D}_l^{neg}$	The set of negative POIs
$C(l_i)$	Hierarchical category set for POI l_i
$\mathcal{D}_{c^j}$	The set of categories at layer j
τ_i	Timestamp
$\langle u, l_i, \tau_i, C(l_i)\rangle$	Check-in activity of user u
s_u	Sequence of check-in activities of user u in a day
$\mathcal{S}$	The set of total check-in sequences for all users
W	Size of sliding window
N	Number of negative samples
d	Embedding size
$\mathbf{v}$	Embedding vectors
$\mathcal{L}$	Objective function

3.2 Problem Formulation

Before diving into the investigated problem, we first introduce some definitions utilized in this work for ease of understanding. We list some important notations in Table 3.4.

Definition 5 ***(Check-in Activity)*** *A check-in activity is denoted by a* $\langle u, l_i, \tau_i, C(l_i)\rangle$ *tuple, which means that user* u *visits POI* l_i *at time* τ_i*. Each POI is geocoded by latitude and longitude, and associated with a well-defined hierarchical category set from bottom to top* $C(l_i) = \{c_{l_i}^1, c_{l_i}^2, \ldots\}$*, where* $c_{l_i}^j$ *is the associated category of POI* l_i *at layer* j *in the category hierarchy* $\mathcal{H}$*. As an example in Fig. 1.2,* $C(l_1) = \{Chinese\ Restaurant, Asia\ Restaurant\}$*.*

Definition 6 ***(Check-in Sequence)*** *For each user, we sort all of her check-ins chronologically, and then divide them into different sequences by day according to [69]. A check-in sequence for user* u *is defined as:* $s_u = \{\langle u, l_1, \tau_1, C(l_1)\rangle, \langle u, l_2, \tau_2, C(l_2)\rangle, \ldots, \langle u, l_n, \tau_n, C(l_n)\rangle\}$*, where* τ_1 *to* τ_n *belong to the same day.*

Given the above definitions, we formally introduce our investigated research problem, that is to recommend the next individual POI with uncertain check-ins at collective POIs, as follows:

Definition 7 ***(Next Individual POI Recommendation with Uncertain Check-ins)*** *Given the current check-in activity $\langle u, l_i, \tau_i, C(l_i)\rangle$ of user u within s_u, our goal is to generate a recommendation list with top-K POIs that user u would be interested in at time τ_{i+1}. Two cases need to be taken into consideration: (1) if the POIs in the list are all individual POIs, it's the final recommendation; and (2) if there is any collective POI in the list, we need to further recommend precise individual POIs inside the associated collective POI.*

3.3 The HCT Framework

To achieve the goal of next individual POI recommendation with uncertain check-ins, we propose a novel hierarchical category transition (HCT) framework based on representation learning techniques. This is motivated by studies in [57, 69], which learn the POI representations through Skip-gram [66] using the POI transition. Different from these conventional methods, we extend Skip-gram to model the category transitions at different layers of CHs as well as the hierarchical dependencies between POIs (categories) and categories. In this way, we can leverage the hierarchical category transition of certain check-ins (outside collective POIs) to infer preferred categories for uncertain check-ins (inside collective POIs), thus comprehensively modeling users' preference transitions. Besides, the hierarchical dependencies between POIs and categories assist in resolving the cold start issue, that is, recommending individual POIs inside collective POIs. The proposed HCT is mainly composed of three modules: data, training and prediction, as depicted by Fig. 3.7. For ease of understanding, we will use the example in Fig. 3.7 to demonstrate each module in the following.

3.3.1 Data Module

As shown in Fig. 3.7, all the POIs $l_1, l_2, \cdots, l_7, \cdots$ (either individual or collective) are characterized by a two-layer category hierarchy $\mathcal{H}$, where l_1, l_2, l_4, l_6, l_7 are individual

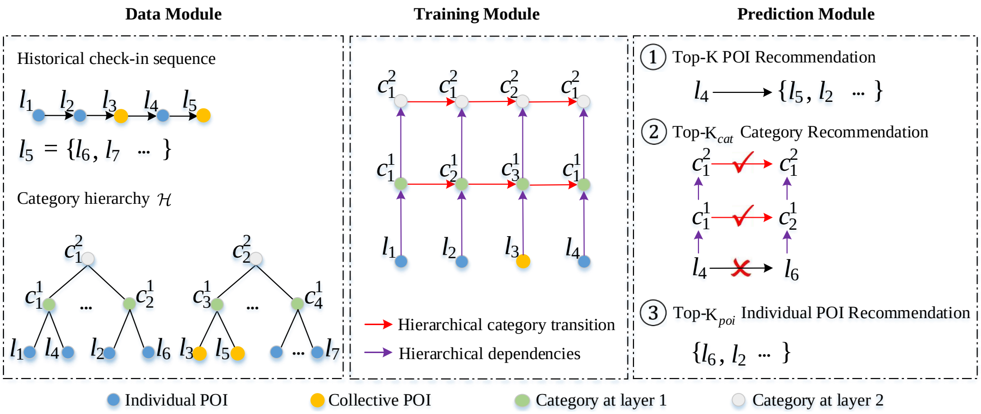


Fig. 3.7: An example to illustrate the overview framework of the proposed HCT, which is composed of three modules: Data Module, Training Module and Prediction Module. The blue (e.g., l_1) and yellow (e.g., l_3) circles represent the individual POI and the collective POI, respectively; the green (e.g., c_1^1) and white (e.g., c_1^2) circles separately represent the first-layer and the second-layer categories that a POI (e.g., l_1) belongs to. The red arrow (e.g., $c_1^1 \to c_2^1 \to c_3^1 \to c_1^1$) stands for the category transition at different layers, while the purple arrow (e.g., $l_1 \to c_1^1 \to c_1^2$) refers to the dependencies between POIs and categories as well as categories at different layers of $\mathcal{H}$. The correct mark indicates that HCT is able to infer the next possible category (e.g., c_2^1/c_1^2) that user u will visit in the collective POI (e.g., l_5) based on her current visited category (e.g., c_1^1/c_1^2); contrarily, the cross mark suggests that HCT cannot directly infer the next possible individual POI (e.g., l_6) in the collective POI (e.g., l_5) for user u based on her current visited POI (e.g., l_4).

POIs; l_3, l_5 are collective POIs; l_6, l_7 are individual POIs inside collective POI l_5. The historical check-in sequence of user u is represented by $l_1 \to l_2 \to l_3 \to l_4 \to l_5$, which is split into two parts: training and test. In particular, we take $l_1 \to l_2 \to l_3 \to l_4$ as the training data, which is then converted into a two-layer hierarchical category transition to help model user check-in preference; and l_5 is the test data, which is the ground-truth for evaluating the generated Top-K recommendations in Prediction Module. As HCT adopts the hierarchical dependencies to help characterize check-in activities, the training sequence ($l_1 \to l_2 \to l_3 \to l_4$) is thus converted to a two-layer category transition by mapping each POI in the sequence into their corresponding categories in the category hierarchy $\mathcal{H}$, that is $c_1^1 \to c_2^1 \to c_3^1 \to c_1^1$ at the first layer and $c_1^2 \to c_1^2 \to c_2^2 \to c_1^2$ at the second layer. Note that, both the two-layer category transition and the hierarchical dependencies between POIs and categories will be as input for the Training Module.

3.3.2 Training Module

In the training module, we extend the Skip-gram [66] to model both the hierarchical dependencies between POIs (categories) and categories, and the hierarchical category transition, thus learning better representations of POIs and categories for high-quality next POI recommendation. The Skip-gram (Word2Vec) model is originated from the natural language processing (NLP) domain, which is an efficient method for learning distributed vector representations for words by capturing a large number of precise syntactic and semantic word relationships [66].

Object Function. Basically, HCT attempts to learn the representations of both POIs and categories by maximizing the following objective function:

$$\mathcal{L}(\mathcal{S}) = \sum_{s_u \in \mathcal{S}} \sum_{\langle l_i, C(l_i)\rangle \in s_u} \sum_{0<w\leq W} \log P(\langle l_{i+w}, C(l_{i+w})\rangle | \langle l_i, C(l_i)\rangle), \tag{3.1}$$

where $\mathcal{S}$ is the set of total check-in sequences for all users; for ease of presentation, hereafter we simplify $s_u = \{\langle u, l_1, \tau_1, C(l_1)\rangle, \langle u, l_2, \tau_2, C(l_2)\rangle, \ldots, \langle u, l_n, \tau_n, C(l_n)\rangle\}$ by only remaining the POI and hierarchical category set, that is, $s_u = \{\langle l_1, C(l_1)\rangle, \langle l_2, C(l_2)\rangle, \ldots,$ $\langle l_n, C(l_n)\rangle\}$; W controls the size of the sliding window; $P(\cdot)$ denotes the probability that a user will visit l_{i+w} belonging to $C(l_{i+w})$ given her current location l_i belonging to $C(l_i)$, which is formulated by considering the hierarchical dependencies and category transitions as follows:

$$P(\langle l_{i+w}, C(l_{i+w})\rangle | \langle l_i, C(l_i)\rangle) = \underbrace{P(l_i | C(l_i))}_{①} \underbrace{P(C(l_{i+w}) | C(l_i))}_{②} \underbrace{P(l_{i+w} | C(l_{i+w}))}_{③}, \tag{3.2}$$

where $P(l_i|C(l_i))$ denotes the probabilistic dependency of POI l_i to its associated hierarchical categories $C(l_i) = \{c^1_{l_i}, c^2_{l_i}, \cdots\}$, that is, the probability of visiting POI l_i for users given the preferred categories $C(l_i)$; $P(C(l_{i+w})|C(l_i))$ denotes the probability of visiting POI belonging to $C(l_{i+w})$ given the current POI belonging to $C(l_i)$. Therefore, $P(l_i|C(l_i))$ and $P(C(l_{i+w})|C(l_i))$ respectively model the hierarchical dependencies, and the hierarchical category transitions, as depicted by Fig. 3.8(a). Hence, HCT fuses two components: (1) hierarchical dependencies for better representations of POIs and categories as well as better semantic relatedness measurement; and (2) hierarchical category transitions for

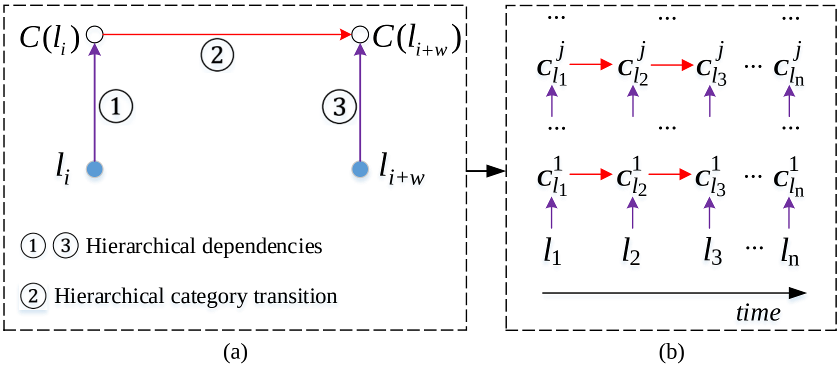


Fig. 3.8: The illustration of modeling the hierarchical dependencies and category transitions.

better multi-granularity user preference transition learning. Next, we will elaborate the two components, respectively.

Modeling Hierarchical Dependencies. Given the current check-in activity $\langle l_i, C(l_i)\rangle$, where $C(l_i) = \{c^1_{l_i}, c^2_{l_i}, \cdots, c^j_{l_i}, \cdots\}$, HCT attempts to predict the probabilistic dependency of l_i to its associated category at the first layer ($c^1_{l_i}$) as well as that of category at j-th layer ($c^j_{l_i}$) to category at $(j+1)$-th layer ($c^{j+1}_{l_i}$), as depicted in Fig. 3.8(b). They can be calculated by the following soft-max functions:

$$P(l_i|c^1_{l_i}) = \frac{\exp(\mathbf{v}^\top_{l_i} \cdot \mathbf{v}_{c^1_{l_i}})}{\sum_{l_{i'} \in \mathcal{D}_l} \exp(\mathbf{v}^\top_{l_{i'}} \cdot \mathbf{v}_{c^1_{l_i}})}, \tag{3.3}$$

$$P(c^j_{l_i}|c^{j+1}_{l_i}) = \frac{\exp(\mathbf{v}^\top_{c^j_{l_i}} \cdot \mathbf{v}_{c^{j+1}_{l_i}})}{\sum_{c^j_{l_{i'}} \in \mathcal{D}_{c^j}} \exp(\mathbf{v}^\top_{c^j_{l_{i'}}} \cdot \mathbf{v}_{c^{j+1}_{l_i}})}, \tag{3.4}$$

where POI l_i and category $c^j_{l_i}$ are represented by low-dimensional continuous vectors $\mathbf{v}_{l_i} \in \mathbb{R}^d$ and $\mathbf{v}_{c^j_{l_i}} \in \mathbb{R}^d$ in the same space, respectively; and d is the dimension; $l_{i'} \in \mathcal{D}_l$ is the negative POI that does not belong to $c^1_{l_i}$; $c^j_{l_{i'}} \in \mathcal{D}_{c^j}$ is the negative category at layer j that does not belong to $c^{j+1}_{l_i}$; $\mathcal{D}_l$ is the set of all POIs; $\mathcal{D}_{c^j}$ is the set of all categories at layer j.

Modeling Hierarchical Category Transition. We simultaneously model the hierarchical category transition as shown in Fig. 3.8(b). With the CH, the check-in sequences of POIs are mapped into category transitions at different layers of the CH. HCT attempts to predict the probability of category $c^j_{l_{i+w}}$, given category $c^j_{l_i}$ of the current POI l_i, which is calculated by:

$$P(c^j_{l_{i+w}}|c^j_{l_i}) = \frac{\exp(\mathbf{v}^{\top}_{c^j_{l_{i+w}}} \cdot \mathbf{v}_{c^j_{l_i}})}{\sum_{c^j_{l_{i'}} \in \mathcal{D}_{c^j}} \exp(\mathbf{v}^{\top}_{c^j_{l_{i'}}} \cdot \mathbf{v}_{c^j_{l_i}})}. \tag{3.5}$$

where $c^j_{l_{i'}}$ is the negative category that does not co-occur with category $c^j_{l_i}$. By considering the hierarchical category transition over the certain check-ins, we are able to infer the possible categories of those uncertain check-ins, thus can address the challenge of modeling incomplete check-in transition. Besides, it also facilitates to learn user preference in different granularity. For instance, the category transition at the higher layer describes more general user preference transition. If a certain category transition at the lower layer is unavailable, we may turn to category transition at the higher layer.

Modeling User Preference. Inspired by [96], the proposed HCT additionally models the dynamic user preference by considering her recently visited POIs and the associated hierarchical category sets. Specifically, we formulate the dynamic user representation by incorporating the representations of POIs (visited before current time τ) as well as the associated hierarchical category sets, given by:

$$\mathbf{v}_u = \sum_{\langle u, l_i, \tau_i, C(l_i)\rangle \in s_u \bigcap (\tau_i < \tau)} \exp^{-(\tau - \tau_i)} \cdot (\mathbf{v}_{l_i} + \sum_{c^j_{l_i} \in C(l_i)} \mathbf{v}_{c^j_{l_i}}), \tag{3.6}$$

where $\mathbf{v}_u$ is the dynamic representation of user u; $\langle u, l_i, \tau_i, C(l_i)\rangle$ is the check-in activity in s_u before time τ.

3.3.3 Prediction Module

In the prediction module, HCT consists of three steps to achieve the goal of next individual POI recommendation with uncertain check-ins, as depicted by the Prediction Module in Fig. 3.7: (1) top-K POI recommendation, (2) top-K_{cat} category recommendation and (3) top-K_{poi} individual POI recommendation. We will introduce them step by step.

Top-K POI Recommendation. Different from traditional category transition based methods first predicting the next preferred categories, we directly generate a list with top-K (either individual or collective) POIs given the current location in the first step, as we believe that the POI transition patterns would be implicitly captured by considering the hierarchical dependencies between POIs and categories in the training module. The category transition is mainly used in the second step, namely, Top-K_{cat} category recommendation. Therefore, given a user u with her currently visited POI l_i (e.g., l_4), we compute the ranking score for each POI l_k (e.g., l_2, l_5) in the candidate pool[5] by considering both user's dynamic preference and the correlation of POIs,

$$r(u, l_k) = \mathbf{v}_u^\top \cdot \mathbf{v}_{l_k} + \mathbf{v}_{l_i}^\top \cdot \mathbf{v}_{l_k}, \tag{3.7}$$

where the first term ($\mathbf{v}_u^\top \cdot \mathbf{v}_{l_k}$) denotes user u's preference towards POI l_k, and the second term ($\mathbf{v}_{l_i}^\top \cdot \mathbf{v}_{l_k}$) denotes the correlation between candidate POI l_k (either individual or collective) with the current POI l_i.

Based on the above equation, we select top-K POIs with highest ranking scores as the initial recommendation list. Note that if the top-K POIs are all individual POIs, then it is the final recommendation list. Otherwise, if there are collective POIs (e.g., l_5) inside the list, we only remain no more than Top-K_{col} ($K_{col} = 2$) collective POIs as candidates by their respective ranking scores in the top-K list, as our data analysis in Section 3.1 shows that most check-in sequences include less than three collective POIs. For each candidate collective POI l_k (e.g., l_5) inside the list, we further utilize a two-step way to recommend precise individual POIs inside l_k, namely top-K_{cat} category recommendation and top-K_{poi} individual POI recommendation, as introduced next.

Top-K_{cat} Category Recommendation. Since the individual POI l_g (e.g., l_6, l_7) inside the collective POI l_k (e.g., l_5) cannot be observed in users' check-in sequences, HCT cannot learn the representations for these individual POIs, thus cannot directly calculate the POI transition probability. That is to say, HCT is not able to directly predict next possible individual POI (e.g., l_6) in the collective POI (e.g., l_5) for user u based on her current visited POI (e.g, l_4), as illustrated by the cross mark between l_4 and l_6 in the Prediction Module of Fig. 3.7. In contrast, HCT can learn the representations of categories of these individual POIs inside collective POIs through the observed certain category transition, thus would calculate the category transition probability. Hence, we adopt the hierarchical category transition of user u to help infer her preferred categories

[5]The candidate pool for user u refers to all the POIs excluding her current POI l_i.

inside collective POI l_k (e.g., l_5) based on her current visited category, as illustrated by the correct marks between $c_1^1(c_1^2)$ and $c_2^1(c_1^2)$ in the Prediction Module of Fig. 3.7.

Specifically, given the hierarchical category set of her current POI l_i: $C(l_i) = \{c^1_{l_i}, c^2_{l_i}, \ldots\}$ (e.g., $C(l_4) = \{c_1^1, c_1^2\}$), we first predict possible hierarchical category sets inside l_k that user may prefer. For each candidate category set $C(l_g) = \{c^1_{l_g}, c^2_{l_g}, \ldots\}$[6] (e.g., $C(l_6) = \{c_2^1, c_1^2\}$), where l_g is inside the collective POI l_k, we compute the ranking score by considering user's dynamic preference and the hierarchical category transition, given by,

$$r'(u, C(l_g)) = \mathbf{v}_u^\top \cdot \sum_{c^j_{l_g} \in C(l_g)} \alpha_j \cdot \mathbf{v}_{c^j_{l_g}} + \sum_{c^j_{l_i} \in C(l_i), c^j_{l_g} \in C(l_g)} \alpha_j \cdot \mathbf{v}^\top_{c^j_{l_i}} \cdot \mathbf{v}_{c^j_{l_g}}, \tag{3.8}$$

where α_j controls the importance of category at layer j, as we believe that categories at different layers have different impacts on characterizing the interactions between users and POIs. $\sum_{j=1}^{N_\mathcal{H}} \alpha_j = 1$, where $N_\mathcal{H}$ is the number of layers of category hierarchy $\mathcal{H}$. We then predict top-K_{cat} ($K_{cat} = 2$) preferred hierarchical category sets within l_k, as our data analysis in Section 3.1 demonstrates that more than 95% of check-ins at collective POIs are associated with no more than two categories. We further recommend individual POIs inside l_k bounded to these categories to user u, as we introduce next.

Top-K_{poi} Individual POI Recommendation. For each preferred hierarchical category set (e.g., $C(l_g)$), there may be a number of individual POIs (e.g., l_m, l_n) characterized by this set, that is, $C(l_g) = C(l_m) = C(l_n)$. To select preferred individual POIs under certain category, most of the conventional approaches [9] consider the geographical influence between candidate POIs and current POI, and recommend candidate POIs with short distance to current POI. However, it is not suitable for our scenario, since individual POIs (e.g., $l_g, l_m, l_n, \cdots$), that are inside the collective POI (e.g., $l_k = \{l_g, l_m, l_n, \cdots\}$) and characterized by the same hierarchical category set (e.g., $C(l_g) = C(l_m) = C(l_n)$), almost have the same distance to the current POI l_i. Besides, we cannot access other information (e.g., check-in popularity, ratings) for these individual POIs except the geographical information, as they cannot be observed in the check-in data.

Therefore, we propose incorporating the local geographical influence to help select individual POIs under a certain category set (e.g., $C(l_g)$). Specifically, we rank all the

[6] There may be a number of individual POIs inside l_k that are characterized by the same hierarchical category set as l_g. We only consider all unique hierarchical category sets.

Table 3.5: Update Rules for HCT.

$$
\begin{aligned}
\mathbf{v}_{l_i} &\leftarrow \mathbf{v}_{l_i} + \eta(1 - \sigma(\mathbf{v}_{l_i}^{\top} \cdot \mathbf{v}_{c^j_{l_i}}))\mathbf{v}_{c^j_{l_i}} \\
\mathbf{v}_{l_{i+w}} &\leftarrow \mathbf{v}_{l_{i+w}} + \eta(1 - \sigma(\mathbf{v}_{l_{i+w}}^{\top} \cdot \mathbf{v}_{c^j_{l_{i+w}}}))\mathbf{v}_{c^j_{l_{i+w}}} \\
\mathbf{v}_{c^j_{l_i}} &\leftarrow \mathbf{v}_{c^j_{l_i}} + \eta\big[(1 - \sigma(\mathbf{v}_{l_i}^{\top} \cdot \mathbf{v}_{c^j_{l_i}}))\mathbf{v}_{l_i} + (1 - \sigma(\mathbf{v}_{c^j_{l_i}}^{\top} \cdot \mathbf{v}_{c^{j+1}_{l_i}}))\mathbf{v}_{c^{j+1}_{l_i}} \\
&\qquad + (1 - \sigma(\mathbf{v}_{c^j_{l_{i+w}}}^{\top} \cdot \mathbf{v}_{c^j_{l_i}}))\mathbf{v}_{c^j_{l_{i+w}}}\big] \\
\mathbf{v}_{c^j_{l_{i+w}}} &\leftarrow \mathbf{v}_{c^j_{l_{i+w}}} + \eta\big[(1 - \sigma(\mathbf{v}_{l_{i+w}}^{\top} \cdot \mathbf{v}_{c^j_{l_{i+w}}}))\mathbf{v}_{l_{i+w}} + (1 - \sigma(\mathbf{v}_{c^j_{l_{i+w}}}^{\top} \cdot \mathbf{v}_{c^{j+1}_{l_{i+w}}}))\mathbf{v}_{c^{j+1}_{l_{i+w}}} \\
&\qquad + (1 - \sigma(\mathbf{v}_{c^j_{l_{i+w}}}^{\top} \cdot \mathbf{v}_{c^j_{l_i}}))\mathbf{v}_{c^j_{l_i}}\big] \\
\mathbf{v}_{c^{j+1}_{l_i}} &\leftarrow \mathbf{v}_{c^{j+1}_{l_i}} + \eta\big[(1 - \sigma(\mathbf{v}_{c^j_{l_i}}^{\top} \cdot \mathbf{v}_{c^{j+1}_{l_i}}))\mathbf{v}_{c^j_{l_i}} + (1 - \sigma(\mathbf{v}_{c^{j+1}_{l_{i+w}}}^{\top} \cdot \mathbf{v}_{c^{j+1}_{l_i}}))\mathbf{v}_{c^{j+1}_{l_{i+w}}}\big] \\
\mathbf{v}_{c^{j+1}_{l_{i+w}}} &\leftarrow \mathbf{v}_{c^{j+1}_{l_{i+w}}} + \eta\big[(1 - \sigma(\mathbf{v}_{c^j_{l_{i+w}}}^{\top} \cdot \mathbf{v}_{c^{j+1}_{l_{i+w}}}))\mathbf{v}_{c^j_{l_{i+w}}} + (1 - \sigma(\mathbf{v}_{c^{j+1}_{l_{i+w}}}^{\top} \cdot \mathbf{v}_{c^{j+1}_{l_i}}))\mathbf{v}_{c^{j+1}_{l_i}}\big]
\end{aligned} \tag{3.9}
$$

$$
\begin{aligned}
\mathbf{v}_{l_{i'}} &\leftarrow \mathbf{v}_{l_{i'}} - \eta\big[\sigma(\mathbf{v}_{l_{i'}}^{\top} \cdot \mathbf{v}_{c^j_{l_i}})\mathbf{v}_{c^j_{l_i}} + \sigma(\mathbf{v}_{l_{i'}}^{\top} \cdot \mathbf{v}_{c^j_{l_i}})\mathbf{v}_{c^j_{l_{i+w}}}\big] \\
\mathbf{v}_{c^j_{l_{i'}}} &\leftarrow \mathbf{v}_{c^j_{l_{i'}}} - \eta\big[\sigma(\mathbf{v}_{c^j_{l_{i'}}}^{\top} \cdot \mathbf{v}_{c^{j+1}_{l_i}})\mathbf{v}_{c^{j+1}_{l_i}} + \sigma(\mathbf{v}_{c^j_{l_{i'}}}^{\top} \cdot \mathbf{v}_{c^{j+1}_{l_{i+w}}})\mathbf{v}_{c^{j+1}_{l_{i+w}}}\big] \\
\mathbf{v}_{c^{j+1}_{l_{i'}}} &\leftarrow \mathbf{v}_{c^{j+1}_{l_{i'}}} - \eta\sigma(\mathbf{v}_{c^{j+1}_{l_{i'}}}^{\top} \cdot \mathbf{v}_{c^{j+1}_{l_i}})\mathbf{v}_{c^{j+1}_{l_i}}
\end{aligned} \tag{3.10}
$$

POIs characterized by $C(l_g)$ within l_k based on their distance to the geographical center of $c^1_{l_g}$ in an ascending order, and then select the top-K_{poi} ($K_{poi} = 2$) individual POIs, as indicated by our data analysis in Section 3.1 that more than 80% of check-ins at collective POIs have no more than 2 visited individual POIs under each category. The final recommendation list is thus composed of both individual POIs (e.g., l_2) in the initial Top-K recommendation list as well as those (e.g., l_6) in the selected collective POIs where the maximum number should be $K_{col} \times K_{cat} \times K_{poi}$.

3.3.4 Model Optimization and Complexity Analysis

Model Optimization. It is computationally expensive to calculate the probabilities in Eq.(3.3)-(3.5), as the denominators need to sum over the entire POI or category set. We, therefore, apply the negative sampling technique [37] to help reduce the training time complexity. We take Eq.(3.3) as an example, which is transformed as below:

$$\log P(l_i|c^1_{l_i}) = \log\sigma(\mathbf{v}^\top_{l_i} \cdot \mathbf{v}_{c^1_{l_i}}) + \sum_{l_{i'} \in \mathcal{D}^{neg}_l}^{N} \log\sigma(-\mathbf{v}^\top_{l_{i'}} \cdot \mathbf{v}_{c^1_{l_i}}), \tag{3.11}$$

where $\sigma(x) = 1/(1 + \exp(-x))$ is the sigmoid function; $\mathcal{D}^{neg}_l$ is the set of negative POIs; $l_{i'}$ is the sampled negative POI that is not associated with category $c^1_{l_i}$; N is the number of negative samples.

The objective function of HCT is to optimize Eq.(3.1). To learn the model, we update the corresponding parameter along the ascending gradient direction, defined as below:

$$\mathbf{v}^{t+1} = \mathbf{v}^t + \eta \frac{\partial \mathcal{L}(\mathbf{v})}{\partial \mathbf{v}}, \tag{3.12}$$

where η is the learning rate. More specifically, given the current and next check-in activities, we get the updating rule by calculating the gradient, defined by Eq.(3.9) in Table 3.5. Simultaneously, we update the negative samples $l_{i'}$, $c^j_{l_{i'}}$ and $c^{j+1}_{l_{i'}}$ by Eq.(3.10) in Table 3.5. The learning algorithm of HCT is summarized in Algorithm 1, which is mainly composed of three modules: data module (lines 2-4), training module (lines 5-11), and prediction module (lines 12-15).

Complexity Analysis. The computational time is mainly taken by evaluating the objective function $\mathcal{L}$ and updating the related variables. The time to compute $\mathcal{L}$ in each iteration is approximate to $O((W \cdot N \cdot d)|\mathcal{D}|)$, where W is the window size; N is the number of negative samples; d is the embedding size; $|\mathcal{D}|$ is the size of all check-in activities. In practice, $W, N, d \ll |\mathcal{D}|$. Therefore, the time complexity of the proposed HCT is linearly with the number of check-in activities $|\mathcal{D}|$, thus being scalable for large-size datasets.

3.4 Experiments and Analysis

We conduct extensive experiments on the three newly-constructed POI datasets to evaluate the effectiveness of our proposed hierarchical category transition (HCT) framework.

3.4.1 Experimental Setup

Datasets. Our experiments are conducted on the three newly-constructed real-world datasets described in the Section 3.1. We divide the datasets into training set and test

Algorithm 1: The Procedure of HCT.

Input: $\mathcal{S}, \mathcal{H}, d, \eta, W, N, iteration$

Output: Final individual POI recommendation list

Initialize: $\mathbf{v}$ with normal distribution;

// Data Module

foreach $u \in \mathcal{U}$ **do**

  Split check-in sequences by day;

  Map daily check-in sequences into category transition by $\mathcal{H}$;

// Training Module

for $t = 1; t \leq iteration; t{+}{+}$ **do**

  foreach $s_u \in S$ **do**

    foreach $\langle l_i, C(l_i)\rangle \in s_u$ **do**

      // $C(l_i) = \{c_i^1, c_i^2, \ldots\}$

      foreach $\langle l_{i+w}, C(l_{i+w})\rangle \in s_u$ **do**

        // $C(l_{i+w}) = \{c_{i+w}^1, c_{i+w}^2, \ldots\}$

        Update parameters according to Eq. (3.9);

        foreach $l_{i'} \in \mathcal{D}_l^{neg}, c_{l_{i'}}^j \in \mathcal{D}_{c^j}^{neg}, c_{l_{i'}}^{j+1} \in \mathcal{D}_{c^{j+1}}^{neg}$ **do**

          Update parameters according to Eq. (3.10);

// Prediction Module

foreach $u \in \mathcal{U}$ **do**

  // ① Top-K POI Recommendation

  Calculate the score according to Eq. (3.7) and select top-K POI;

  // ② Top-K_{cat} Category Recommendation

  Calculate the score according to Eq. (3.8) and select top-K_{cat} categories;

  // ③ Top-K_{poi} Individual POI Recommendation

  Rank POIs by geographical distance and select top-K_{poi} individual POIs.

set based on the timestamp, where the earlier check-ins are as the training set, and the most recent check-ins are as the test set. In particular, we adopt 90% of each dataset as the training set, and the rest as the test set, as in [97]. Following [4], given a user and her current location, we predict her next check-in, and use all her next check-ins in the test set within the successive $\tau = 6$ hours as the ground-truth.

Comparison Methods. To evaluate the effectiveness of our proposed **HCT** method, the following state-of-the-art algorithms are compared: (1) **Most Popular** (MP): it recommends the most popular POIs to each user; (2) **BPRMF** [98]: it is the classic Bayesian personalized ranking model based on matrix factorization; (3) **ME** [4]: it is the metric embedding model which learns the user's sequential transition for next POI recommendation; (4) **LBPR** [9]: it is the listwise Bayesian personalized ranking method, which predicts users' preferred next categories first, and then recommend individual POIs based on the geographical influence; and (5) **SG** [57]: it is the recent Skip-gram based POI embedding method which models user's sequential transition for next POI recommendation.

All the comparison methods are not directly suitable for our investigated problem, i.e., recommending individual POIs with uncertain check-ins at collective POIs. To this end, we adapt these methods by following the same procedure as our HCT. First, we generate a top-K POI recommendation list for each method. If there is no collective POI inside the list, we treat this list as the final recommendation results. Second, if there are collective POIs inside the recommendation list, we also only remain $K_{col} = 2$ collective POIs as candidates by their respective ranking scores in the top-K list. Third, for each collective POI, we select the $K_{cat} = 2$ most popular categories within this collective POI. The popularity is calculated based on all users' certain check-in behavior. Lastly, we further select $K_{poi} = 2$ individual POIs under each selected category by using the same strategy as HCT does. The final recommendation is thus composed of both individual POIs in the initial top-K list, as well as those in the selected collective POIs, where the maximum number should be $K_{col} \times K_{cat} \times K_{poi}$.

Evaluation Metrics. Following [97], we use two standard metrics, namely precision@K and recall@K to evaluate the performance of all the methods. Given the top-K recommended POIs for each user,

$$\text{precision@}K = \frac{|S_{Rec} \bigcap S_{Visited}|}{K} \tag{3.13}$$

$$\text{recall@}K = \frac{|S_{Rec} \bigcap S_{Visited}|}{|S_{Visited}|} \tag{3.14}$$

where S_{Rec} is the set of top-K recommended POIs, and $S_{Visited}$ is the set of actually POIs that the user visited within the successive $\tau = 6$ hours in the test set as in [3, 4]. We study the performance of all comparison methods by averaging the precision and recall of all users. Generally, the higher precision and recall, the better the recommendation performance.

Parameter Settings. We find out the optimal parameter settings for all comparison methods by either empirical study or following suggestions in the original papers. Specifically, we apply a grid search in $\{50, 100, 150, 200, 250, 300\}$ to find out the optimal value for embedding size d; a grid search in $\{0.001, 0.005, 0.01, 0.05, 0.1, 0.5, 1.0\}$ to find out the best settings for learning rate and regularization coefficient; and a grid search in $\{1, 2, 3, 4\}$ to find out the best setting for window size W; $K_{col} = 2, K_{cat} = 2, K_{poi} = 2$; $\tau = 6$; the list size of LBPR is set to 2; for ME, $\alpha = 0.2$ controlling the importance of user preference and sequential influence; for HCT, $d = 200, \eta = 0.001$; $N = 100, 30, 20$ for the number of negative POIs, negative categories at the first and second layer, respectively; a grid search in $[0, 1]$ stepped by 0.1 is applied to find out the best settings for α_j, which controls the importance of category at different layers of the CH.

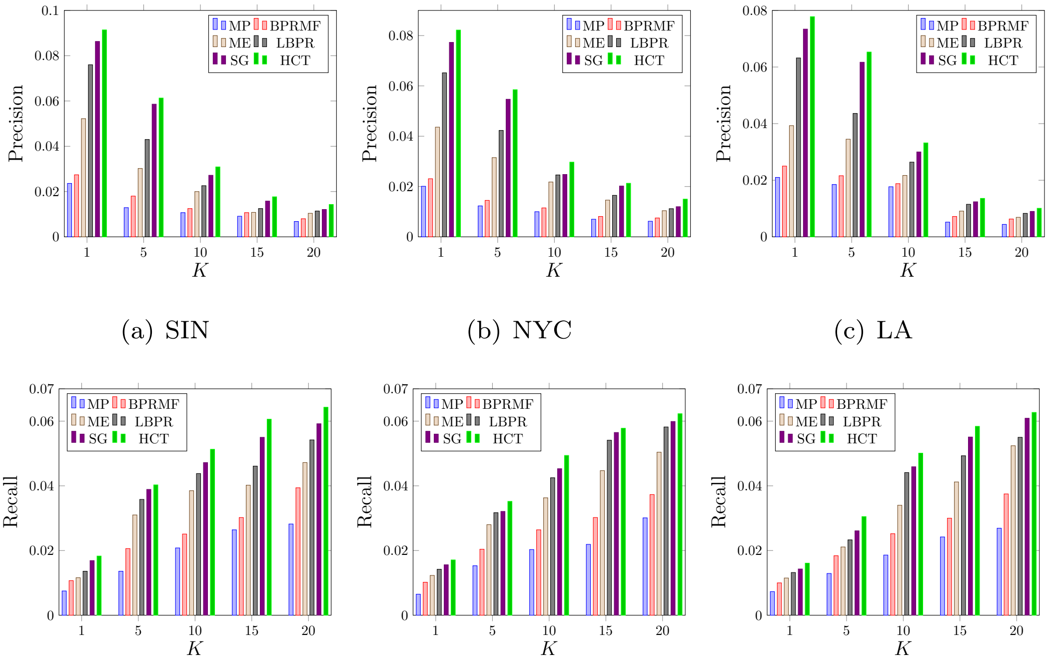
3.4.2 Results and Analysis

To systematically evaluate all the methods, we design an all-around evaluation procedure regarding to the three components in prediction module: (1) top-K *POI recommendation*, (2) top-K_{cat} *category recommendation*, and (3) top-K_{poi} *individual POI recommendation*. Additionally, we further investigate the impacts of hyper-parameters on the final recommendation results, followed by the evaluation of time complexity for all the comparison methods.

3.4.2.1 Results on Top-K POI Recommendation

We first study the performance of all comparisons on Top-K POI recommendation, which may recommend both individual and collective POIs. Note that, our data analysis in Section 3.1 shows that most check-in sequences include no more than two collective POIs. Therefore, we only remain $K_{col} = 2$ collective POIs in the top-K list. We vary K with values in $\{1, 5, 10, 15, 20\}$, and the results on the three datasets are depicted by Fig. 3.9. From the results, several interesting findings are noted as follows.

(1) As the value of K increases, the precision of all methods decreases, while the recall increases. These results are in accordance with all existing next POI recommendation approaches [53, 69]. (2) The popular-based method MP is generally outperformed by other methods, implying the superiority of latent factor models, metric embedding



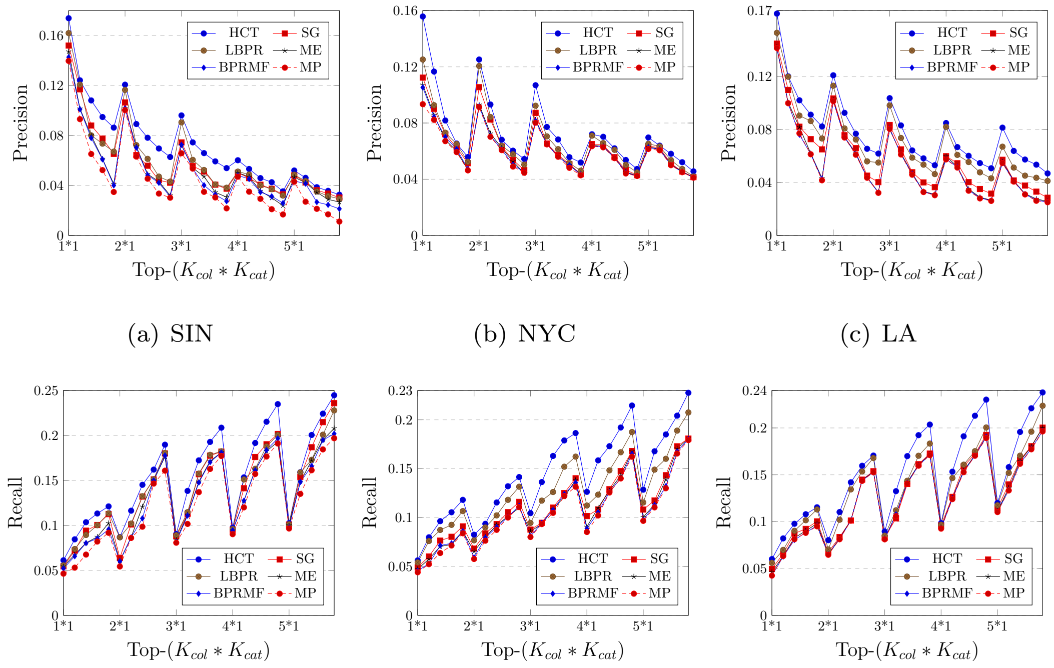
(a) SIN (b) NYC (c) LA

(d) SIN (e) NYC (f) LA

Fig. 3.9: Performance comparison for top-K POI recommendation on the three datasets.

models and representation learning models on user preference inference. (3) The latent factor model based methods LBPR outperforms BPRMF and metric embedding model ME, which indicates that incorporating category transition is helpful for better next POI recommendation. (4) The representation learning model based methods (SG and HCT) perform better than the latent factor models (LBPR and BPRMF) and metric embedding model (ME). This helps demonstrate the effectiveness of representation learning models on achieving better recommendation accuracy, which is also in accordance with the conclusion in early study [5, 53, 68]. (5) The proposed HCT consistently outperforms all the other comparisons across all the metrics on the three datasets, and the improvements are significant with paired t-test (p-value < 0.01). This suggests that, considering the hierarchical category transition as well as the hierarchical dependencies between POIs and categories is better than modeling check-in sequential transition directly.

Given the top-K_{cat} recommended categories for each selected collective POI, we finally recommend individual POIs under each preferred category. Our data analysis in Section 3.1 suggests that more than 80% of check-ins at collective POIs have no more than 2 visited individual POIs under each category. We vary the number of individual



(a) SIN (b) NYC (c) LA

(d) SIN (e) NYC (f) LA

Fig. 3.10: Performance comparison w.r.t. category correctness on the three datasets.

POIs with $K_{poi} = \{1, 2, 3, 4\}$ to investigate its impacts on the final recommendation results, which is composed of both individual POIs in the initial top-K list as well as these in the selected collective POIs. For a fair comparison, we fix $K = 5, K_{col} = 2, K_{cat} = 2$.

The overall performance of all comparisons is summarized in Fig. 3.11. We observe that: (1) similar as the results on top-K POI recommendation and top-K_{cat} category recommendation, all the methods achieve a smaller value w.r.t. precision while a larger value w.r.t. recall with the increasing of K_{poi}; (2) the proposed HCT performs the best among all the comparisons, which implies that the incorporation of hierarchical category transition and category dependencies is effective for resolving our investigated problem. In particular, the final recommendation results are composed of both individual POIs within the initial top-K recommendation list as well as the individual POIs under the preferred categories inside the selected collective POIs. That is to say, the performance is influenced by both top-K POI recommendation and top-K_{cat} category recommendation. The methods (MP, BPRMF, ME, and SG) based on popularity-aware strategy for category recommendation achieve comparable performance in the second step, as shown in Fig. 3.10, while different performance in the final recommendation has been easily

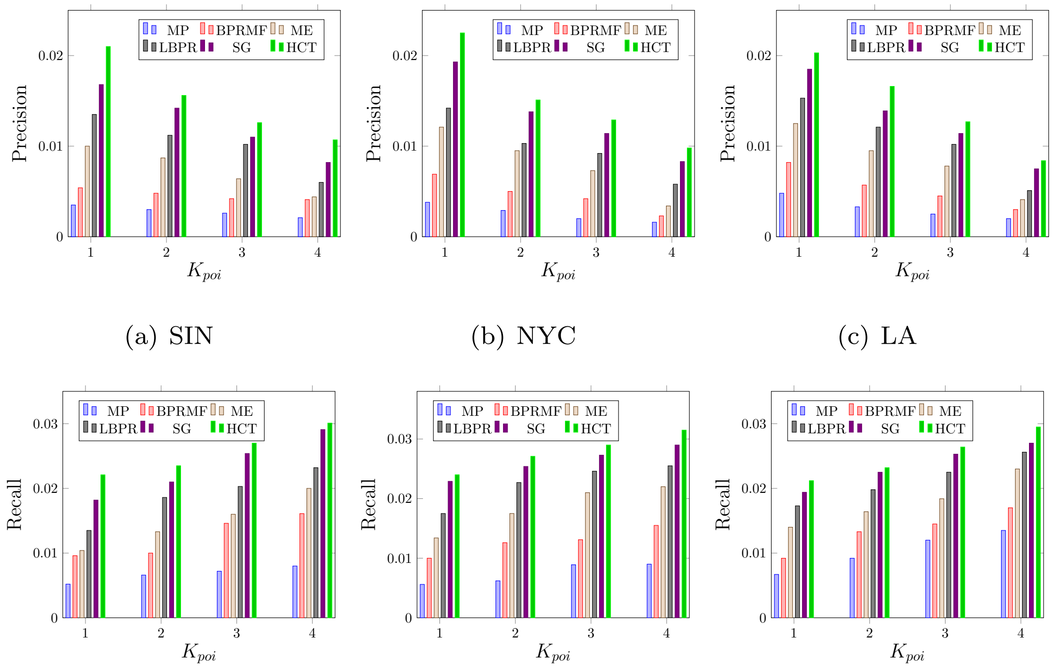


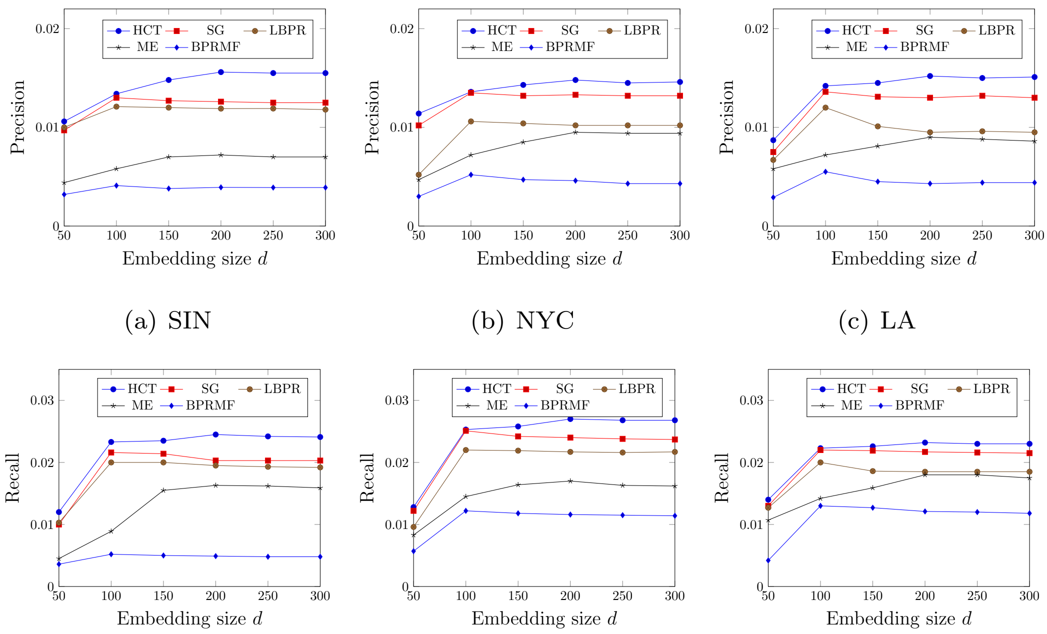
Fig. 3.11: Performance comparison for the Top-K_{poi} individual POI recommendation on three datasets.

observed. This indicates the importance of predicting correct collective POIs in the first step.

3.4.2.2 Effects of Hyper-parameters

We further analyze the effects of hyper-parameters in our experiments, including the embedding size d, the number of window size W, the contribution of categories at different layers of the CH α_j and the data sparsity issue. To achieve a fair comparison, we only vary the value of the investigated parameter with all the other parameters fixed.

Effect of Embedding Size d. To investigate the effect of embedding size d on the final recommendation performance, we vary the value of d in the range of $[50, 300]$ stepped by 50 for BPRMF, ME, LBPR, SG and HCT, as MP is the popularity-based method and unaffected by parameter d. The results on the three datasets are depicted by Fig. 3.12, where we can observe that with the increasing of d, the performance of all the methods improves, and finally keeps stable. The best performance is respectively obtained with $d = 100, 200, 100, 100, 200$ for BPRMF, ME, LBPR, SG, and HCT on the three datasets.



(a) SIN (b) NYC (c) LA

(d) SIN (e) NYC (f) LA

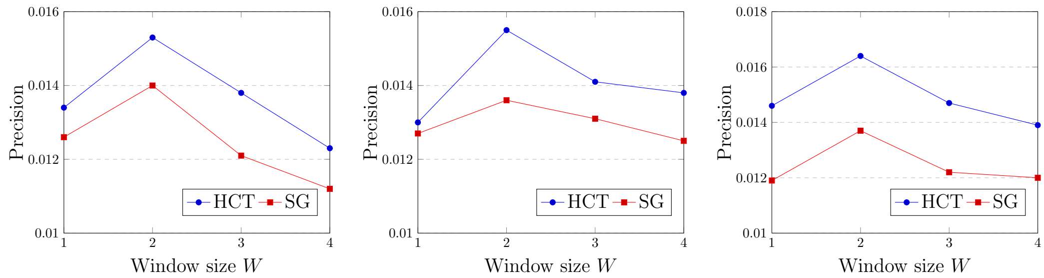
Fig. 3.12: The effect of embedding size d.

Besides, our proposed method HCT generally outperforms other comparison methods with the varying of d on the three datasets.

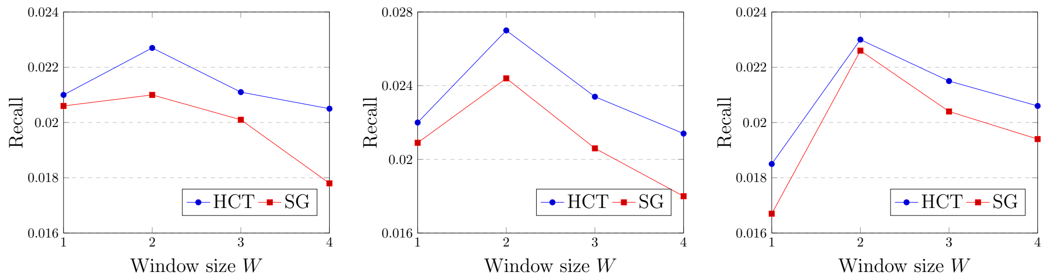
Effect of Window Size W. To investigate the effect of window size W in SG and HTC[7], we vary the value of W in the range of $[1, 4]$ stepped by one. The resutls on the three datasets are shown in Fig. 3.13, where we notice that: (1) our proposed method HCT significantly outperforms SG across different settings of W on the three datasets; (2) the performance of both SG and HCT goes up first and then drops a lot with the increasing of W. The best performance is achieve with $W = 2$ for both SG and HCT on the three datasets.

Effect of Parameter α_j. We investigate the effect of parameter α_j in Eq. (3.8) on the final recommendation performance, which controls the importance of categories at different layers of the hierarchy. As a two-layer category hierarchy is considered in our study, we use α_1 and α_2 to respectively denote the weights of categories at layer 1 and layer 2, where $\alpha_1 + \alpha_2 = 1$. We vary α_2 in the range of $[0, 1]$ with interval 0.1. Fig. 3.14

[7]There is no parameter W in the other comparison methods.



(a) SIN (b) NYC (c) LA



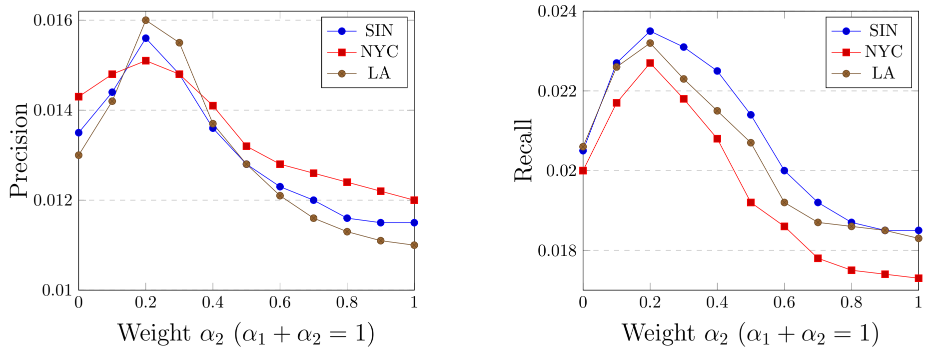
(d) SIN (e) NYC (f) LA

Fig. 3.13: The effect of window size W.

depicts the results on the three datasets, where $\alpha_2 = 0$ ($\alpha_1 = 1$) indicates that only categories at the first layer are taken into consideration; whereas $\alpha_2 = 1$ ($\alpha_1 = 0$) means that only categories at the second layers are incorporated.

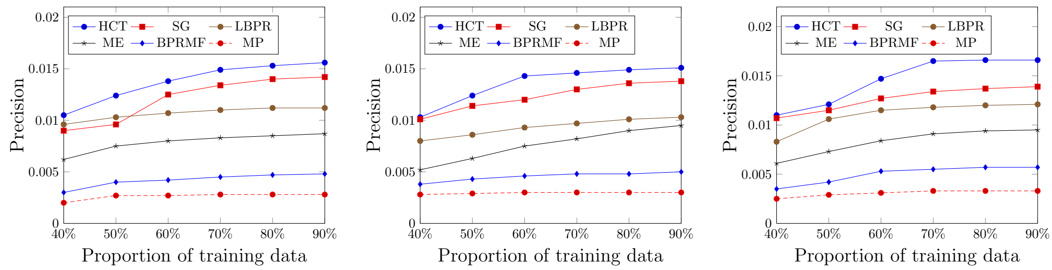
Based on the results, we can observe that: (1) the performance with $\alpha_2 = 0$ is better than that with $\alpha_2 = 1$ suggesting that the categories at lower layer will cover more semantic relatedness than the categories at upper layer in modeling user's preference transition; (2) the best performance is obtained when $\alpha_2 = 0.2$ ($\alpha_1 = 0.8$), which implies that only an appropriate combination of categories at different layers of the hierarchy can help deliver better recommendation performance; and (3) the similar performance variance with the change of α_2 on the three datasets demonstrates the robustness of our proposed HCT.

Effect of Data Sparsity. To investigate the effect of data sparsity on the final recommendation results, we adopt different proportion of training data in the range of [40%, 90%] scaled by 10%, as in [97]. The results are illustrated in Fig. 3.15, where we observe that: (1) MP is unaffected by the size of training data, as there is no training

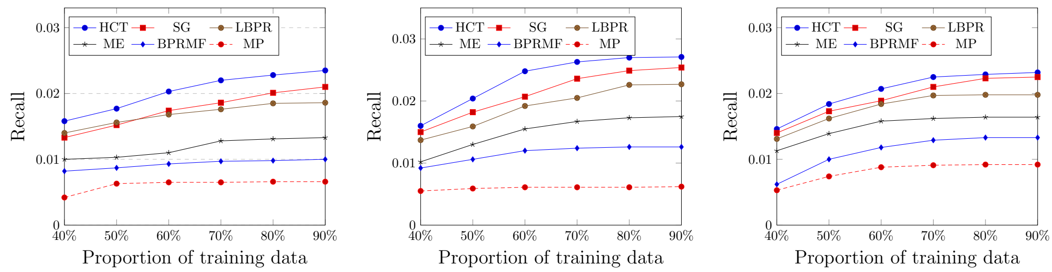


(a) Precision on HCT (b) Recall on HCT

Fig. 3.14: The effect of α_j.



(a) SIN (b) NYC (c) LA



(d) SIN (e) NYC (f) LA

Fig. 3.15: The effect of data sparsity.

process for MP; (2) as the proportion of training data increases, the performance of all the other methods except MP improves gradually, which suggests that sufficient training data can help improve the performance; and (3) the proposed HCT consistently outperforms all the other comparison methods across different proportions of training data on the three datasets, which demonstrates the superiority of HCT on dealing with the data sparsity issue.

3.5 Summary

Category transitions and category hierarchies have proven to be effective in capturing users' preference transitions and relations between categories, respectively. This chapter develops a novel hierarchical category transition (HCT) framework based on representation learning techniques, aiming to resolve a realistic problem in next POI recommendation, that is, recommending next precise individual POI to users with uncertain check-ins. By exploiting: (1) the category transitions at different layers of the CH and (2) the hierarchical dependencies between POIs (categories) and categories in the CH, HCT is capable of capturing users' preference transition patterns over uncertain check-ins in different granularity, and well addressing the cold start issue caused by the uncertain check-ins at collective POIs (individual POIs inside collective POIs are all cold start POIs). Extensive experiments on the three real-world datasets show the superiority of HCT against state-of-the-art algorithms.

Chapter 4

iMTL: Interactive Multi-task Learning Method[1]

In Chapter 3, we first formulate a new research problem of recommending next POIs with uncertain check-ins. Then, we propose HCT by exploiting both effects of the category transitions and hierarchical dependencies to resolve our proposed research problem. This chapter further leverages context features derived from users' historical trajectories for better recommendations. In recent years, studies on the next POI recommendation mainly seek to learn users' transition patterns (i.e., category-level and location-level preference transitions) from trajectories. However, most existing methods suffer from the transition pattern vanishing issue due to the presence of uncertain check-ins (i.e., fuzzy and incomplete sequential behaviors), which hinders the user preference learning and leads to inaccurate prediction. We, therefore, explore users' underlying activities over uncertain check-ins and model the interplay between sequential activities and locations to ease the issue of uncertain check-ins.

In this chapter, we propose a novel interactive multi-task learning (iMTL) framework to better exploit the interplay between activity and location preference. Specifically, iMTL introduces: (1) temporal-aware activity encoder equipped with fuzzy characterization over uncertain check-ins to unveil the latent activity transition patterns; (2) spatial-aware location preference encoder to capture the latent location transition patterns; and (3) task-specific decoder to make use of the learned latent transition patterns

[1]The work in this chapter has been accepted by the 29th International Joint Conference on Artificial Intelligence (IJCAI) [35], 2020.

and enhance both activity and location prediction tasks in an interactive manner. Extensive experiments on the three real-world datasets show the superiority of iMTL against state-of-the-art next POI recommendation approaches.

The remainder of this chapter is organized as follows. In Section 4.1, we first report the data description and analysis. We then illustrate the proposed iMTL framework in Section 4.2. Section 4.3 presents the experimental results, followed by the conclusion in Section 4.4.

4.1 Data Description and Analysis

Although the pioneer work HCT is able to handle the issue of uncertain check-ins in the next POI recommendation, it faces the challenge of selecting individual POIs inside a collective POI when a possible category is predicted, because the other information of such POIs is unavailable except for their categories. Enriching the side information of those POIs is an intuitive way to help ease the cold start issue. Hence, we consider assigning ratings for all POIs by merging Foursquare and Yelp based on their geographical information and categories. In this way, we can only collect three datasets, i.e., Charlotte (CLT), Calgary (CAL) and Phoenix (PHO), which contain rating information. We then group individual POIs into different collective POIs followed by [32]. For instance, individual POIs, e.g., l_6, l_7, l_8, are grouped into a collective POI l_5 as shown in Fig. 1.1. The original check-ins $l_1 \rightarrow l_2 \rightarrow l_6$ are converted to $l_1 \rightarrow l_2 \rightarrow l_5$. Each check-in record is formed as a tuple (u, l, t, c, y, g), meaning that user u visited POI l at time t, where l is associated with category c and POI type y ($y = 0$ denotes individual POI; $y = 1$ refers to collective POI) as well as geocoded by g, i.e., (latitude, longitude). Following [4], we remove users and POIs with fewer than 10 check-ins. Besides, each individual POI is assigned with rating derived from Yelp. The statistics of the newly-constructed datasets are shown in Table 4.1, and three interesting observations are noted by the data analysis.

Observation 1: temporal-aware activities

User's activities, which can be represented by categories of POIs [7], vary with the temporal context [70]. Fig. 4.1(a) shows the ratio of check-ins for the top-4 most popular

Table 4.1: The statistics of three newly-constructed POI datasets.

	# User	# POI	# Check-in	# Category
CLT	1580	1791	20940	239
CAL	301	985	13954	184
PHO	1623	2441	22620	251

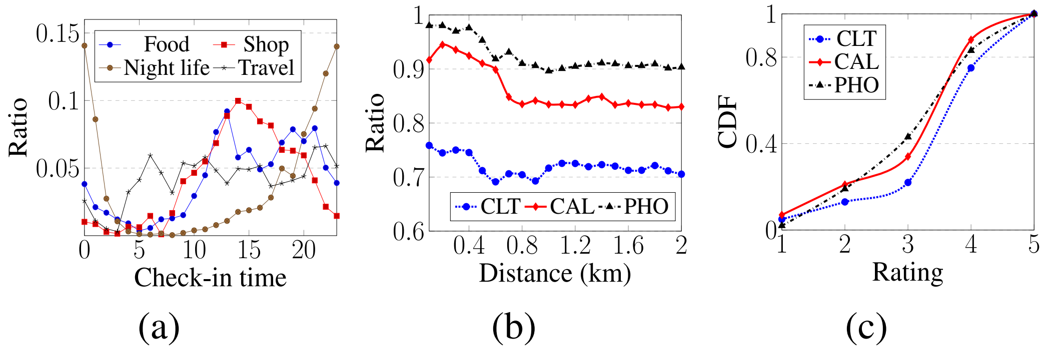


Fig. 4.1: Three observations obtained from the data analysis.

location categories w.r.t the check-in time on CLT[2]. The temporal activity pattern is obvious, e.g., the check-ins of shops occur most often between 10am and 8pm. In contrast, the check-ins of night life start to rise quickly at 7pm and peak at 11pm. These imply that users' activities exhibit the strong temporal pattern, which may significantly impact the recommendation performance.

Observation 2: choice-driven check-ins at collective POIs

Due to the extreme challenge of modeling uncertain check-ins, we attempt to mine underlying factors that affect users' check-in behaviors at collective POIs. Specifically, we explore the consecutive check-ins containing collective POIs within a trajectory. Taking $l_2 \rightarrow l_5$ in Fig. 1.1 as an example, we first search for the POIs (e.g., l_k) near l_5, i.e., $dist(l_k, l_5) \leq \Delta d$, where Δd is the distance threshold. Then we find that Bob visited l_5 instead of l_3 and l_4 (both are the accessible POIs which meet Bob's activity c_1). This is in align with the intuition that users tend to visit collective POIs with more choices (choice-driven), e.g., l_5 contains more individual POIs corresponding to the category c_1. Fig. 4.1(b) shows the different ratio of POIs satisfies the choice-driven over all accessible POIs varying Δd within $2km$ on three datasets. We find that more than 75% uncertain

[2]Due to the similar trends on the three datasets, we thus only show the results on CLT to avoid repeatedly visualize the similar results.

check-ins within $0.4km$ are affected by the scale of POIs (choices) on three datasets. Note that the proportion of cases on CAL and PHO that satisfy the choice-driven property is higher than on CLT.

Obseration 3: rating-driven check-ins at cold start POIs

All individual POIs inside the collective POIs are unobserved in users' check-in records in our scenario, i.e., cold start POIs. It is hard to recommend such POIs even with their geo-location as claimed in HCT [32]. We assign a rating for each individual POI by merging Foursquare and Yelp datasets based on its geographical and category information, as such side information could help ease the cold start issue [15]. We compute the ratio of check-ins over the ratings, and plot the cumulative distribution function (CDF) in Fig. 4.1(c). We find that users prefer individual POIs with higher ratings inside the collective POI. For instance, more than 70% check-ins inside the collective POIs correspond to individual POIs with ratings larger than 3 in CLT.

4.2 Problem Formulation

A trajectory of a user u is his daily historical check-in records, i.e., (u, l, t, c, y, g), in chronological order. We can obtain two types of sequences derived from a trajectory:

- The i-th temporal-aware activity (category) sequence of user u is denoted by a set of activity tuples, i.e., $\mathcal{A}^{u,i} = \{A^u_{t_1}, A^u_{t_2}, \ldots\}$, where $A^u_{t_k} = (c^u_{t_k}, y^u_{t_k}, t^u_k)$, $y^u_{t_k} \in \{0, 1\}$.

- The i-th spatial-aware location sequence of user u is denoted by a set of location tuples, i.e., $\mathcal{L}^{u,i} = \{L^u_{t_1}, L^u_{t_2}, \ldots\}$, where $L^u_{t_k} = (l^u_{t_k}, g^u_{t_k})$.

Given a temporal-aware activity sequence $\mathcal{A}^{u,i}$ and a spatial-aware location sequence $\mathcal{L}^{u,i}$, our goal is to predict user u's next activity $c_{t_{n+1}}$ and location $l_{t_{n+1}}$ at next timestamp t_{n+1}. If $c_{t_{n+1}}$ happens at a collective POI $l_{t_{n+1}}$, we need to further recommend precise individual POIs inside $l_{t_{n+1}}$ given $c_{t_{n+1}}$.

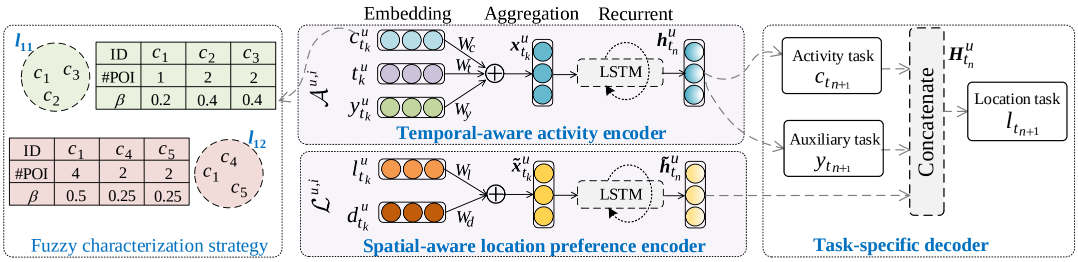


Fig. 4.2: The architecture of the proposed iMTL, which mainly consists of three modules: (1) temporal-aware activity encoder including a fuzzy characterization strategy, (2) spatial-aware location preference encoder, and (3) task-specific decoder. The left subfigure illustrates the fuzzy characterization strategy for the uncertain activity representation based on Observation 2. For instance, collective POI l_{12} contains three categories as shown in the first row of the table; the second row contains the corresponding number of individual POIs under each category within l_{12}; β in the third row is the ratio of #POI to total #POI inside l_{12}.

4.3 The iMTL Framework

The overall architecture of iMTL is outlined in Fig. 4.2. It mainly consists of a two-channel encoder (i.e., temporal-aware activity and spatial-aware location preference encoders) and a task-specific decoder. The two-channel encoder, equipped with embedding, aggregation and recurrent layers, aims to capture the sequential correlations of activity and location preferences. Then the representations encoded by the recurrent layer are utilized in the task-specific decoder to interactively perform three (i.e., the next activity, POI type and POI) prediction tasks.

4.3.1 Temporal-aware Activity Encoder

Given an activity tuple $A^u_{t_k} = (c^u_{t_k}, y^u_{t_k}, t^u_k)$ of user u, $y^u_{t_k} = 1$ represents $c^u_{t_k}$ is an uncertain activity, which happens at collective POI $l^u_{t_k}$ (e.g., Building l_5 in Fig. 1.1). To better represent such an uncertain activity, we propose a fuzzy characterization strategy inspired by Observation 2:

$$\mathbf{c}^u_{t_k} = \sum\nolimits_{j=1}^{M} \beta_j \mathbf{c}_j, \tag{4.1}$$

where M is the total number of categories inside $l^u_{t_k}$; $\mathbf{c}_j \in \mathbb{R}^{D_c}$ is the embedding of j-th category. Since a user prefers a collective POI with more choices given his activity (Observation 2), we thus use β_j to control the possibility of each activity inside $l^u_{t_k}$, which is defined by the scale of individual POIs belong to c_j inside $l^u_{t_k}$ (see Fig. 4.2). Therefore, $\mathbf{c}^u_{t_k}$ can be used to denote the fuzzy representation of user u's uncertain activity, as well as the functionality (e.g., food-oriented) of the collective POI $l^u_{t_k}$. Meanwhile it also well accommodates the scenario where $y^u_{t_k} = 0$, that is, $c^u_{t_k}$ is a certain activity. In that case, $M = 1$, and $\mathbf{c}^u_{t_k}$ is thus the representation of user u's certain activity.

Furthermore, inspired by Observation 1, modeling sequential activities by incorporating temporal contexts is essential to generate accurate user's next activity. Hence, the temporal-aware activity aggregation $\mathbf{x}^u_{t_k}$ is ultimately represented by:

$$\mathbf{x}^u_{t_k} = \mathbf{W}_c \mathbf{c}^u_{t_k} + \mathbf{W}_t \mathbf{t}^u_k + \mathbf{W}_y \mathbf{y}^u_{t_k} + \mathbf{b}, \tag{4.2}$$

where $\mathbf{W}$ is the weight matrix; $\mathbf{b}$ is the bias term; $\mathbf{t}^u_k \in \mathbb{R}^{D_t}$ is the embedding of temporal context with one day mapped into 24 hours; and $\mathbf{y}^u_{t_k} \in \mathbb{R}^{D_y}$ is the embedding of POI type. Subsequently, $\mathbf{x}^u_{t_k}$ is fed into the recurrent layer to infer the hidden state of user u's activity at t_k:

$$\mathbf{h}^u_{t_k} = \mathrm{LSTM}(\mathbf{x}^u_{t_k}, \mathbf{h}^u_{t_{k-1}}), \tag{4.3}$$

where LSTM($\cdot$) captures the sequential correlations of activities, and $\mathbf{h}^u_{t_{k-1}}$ encodes the previous activity until t_{k-1}.

4.3.2 Spatial-aware Location Preference Encoder

As a user's mobility is generally affected by the distance between the current location and the next visiting one [4], the spatial-aware location preference encoder aims to capture sequential location correlations by considering the spatial contexts. Hence, the aggregation $\widetilde{\mathbf{x}}^u_{t_k}$ for each location tuple $L^u_{t_k} = (l^u_{t_k}, g^u_{t_k})$ is defined as:

$$\widetilde{\mathbf{x}}^u_{t_k} = \mathbf{W}_l \mathbf{l}^u_{t_k} + \mathbf{W}_d \mathbf{d}^u_{t_k} + \widetilde{\mathbf{b}}, \tag{4.4}$$

where $\mathbf{l}^u_{t_k} \in \mathbb{R}^{D_l}$ is the embedding of POI $l^u_{t_k}$; and $\mathbf{d}^u_{t_k} \in \mathbb{R}^{D_d}$ is the embedding of the distance interval $d^u_{t_k}$ between $l^u_{t_{k-1}}$ and $l^u_{t_k}$ based on $g^u_{t_{k-1}}$ and $g^u_{t_k}$. Note that we round the distance into integer (e.g., $2.12 \rightarrow 2$), to reduce the number of parameters. $\widetilde{\mathbf{x}}^u_{t_k}$ is then fed into the recurrent layer to infer the hidden state $\widetilde{\mathbf{h}}^u_{t_k}$ of location preference at t_k, formulated as,

$$\widetilde{\mathbf{h}}^u_{t_k} = \mathrm{LSTM}(\widetilde{\mathbf{x}}^u_{t_k}, \widetilde{\mathbf{h}}^u_{t_{k-1}}). \tag{4.5}$$

4.3.3 Task-specific Decoder

The task-specific decoder aims to interactively perform three (i.e., the next activity, POI type and location) prediction tasks based on the latent representations, i.e., $\mathbf{h}_{t_n}^u$ and $\widetilde{\mathbf{h}}_{t_n}^u$, learned from the two-channel encoder.

Activity Prediction with Auxiliary Task. We predict user u's next activity in the dot-product way, and the probability of next possible activity $c_{t_{n+1}}$ at time t_{n+1} is calculated by:

$$r_{t_{n+1},c_{t_{n+1}}}^u = (\mathbf{W}_h^c \mathbf{h}_{t_n}^u)^\top (\mathbf{W}_c \mathbf{c}_{t_{n+1}} + \mathbf{W}_y \mathbf{y}_{t_{n+1}} + \mathbf{W}_t \mathbf{t}_{n+1}^u + \mathbf{b}), \tag{4.6}$$

where the first term $\mathbf{W}_h^c \mathbf{h}_{t_n}^u$ represents the user representation by encoding the dynamic hidden states, and the second term $\mathbf{W}_c \mathbf{c}_{t_{n+1}} + \mathbf{W}_y \mathbf{y}_{t_{n+1}} + \mathbf{W}_t \mathbf{t}_{n+1}^u + \mathbf{b}$ denotes the temporal-aware activity representation.

Although the fuzzy characterization strategy can help mine the user's underlying activity, it may introduce some noise in the process of activity prediction. Inspired by [39], we thus perform the major task (i.e., activity prediction) with the help of the auxiliary task (i.e., POI type prediction). This is mainly because of two reasons: (1) POI type prediction is related to the activity prediction, as it triggers the presence of either certain or uncertain activities[3]; (2) jointly learning both tasks can enhance the model generalization by averaging the inevitable noise introduced by the uncertain activity representation. The POI type prediction task is formulated as below, where σ is the sigmoid function,

$$r_{t_{n+1},y_{t_{n+1}}}^u = \sigma(\mathbf{W}_h^y \, \mathbf{h}_{t_n}^u). \tag{4.7}$$

Location Prediction with Interactive Fashion. Considering the interplay between the user's activity and location visit, that is, the next location check-in is affected by the activity [32], we propose an interactive multi-task learning strategy to generate the next location prediction under the help of activity prediction with auxiliary task. Inspired by [22], we concatenate the latent representations learned in the two-channel encoder together with the predicted results of activity and POI type:

$$\mathbf{H}_{t_n}^u = [\widetilde{\mathbf{h}}_{t_n}^u; \mathbf{h}_{t_n}^u;\ r_{t_{n+1},c_{t_{n+1}}}^u \mathbf{c}_{t_{n+1}};\ r_{t_{n+1},y_{t_{n+1}}}^u \, \mathbf{y}_{t_{n+1}}], \tag{4.8}$$

[3]To accurately predict the activity and POI type greatly helps us locate the user's activity that happens inside a collective POI, rather than search for other individual POIs under the predicted activity.

Algorithm 2: Learning Strategy of iMTL.

Input: $\Theta, \mathcal{A}, \mathcal{L}$, user set $\mathcal{U}$, max_iter, λ, η
Output: iMTL model $\{\Theta\}$
Initialize the set of model parameters Θ;
`// Training instances generation`
foreach $u \in \mathcal{U}$ **do**
  foreach $\mathcal{A}^{u,i} \in \mathcal{A}^u$ *and* $\mathcal{L}^{u,i} \in \mathcal{L}^u$ **do**
    Sample negative activities and locations
`// Parameter Update`
for $iter = 1; iter \leq max\_iter; iter++$ **do**
  foreach $u \in \mathcal{U}$ **do**
    Randomly select a batch of instances
    foreach $\theta \in \Theta$ **do**
      $\theta \leftarrow \theta - \eta * \frac{\partial \mathcal{J}}{\partial \theta}$

Hence, the probability of u's next POI $l_{t_{n+1}}$ at time t_{n+1} is formulated as:

$$r^u_{t_{n+1}, l_{t_{n+1}}} = (\mathbf{W}_H \mathbf{H}^u_{t_n})^\top (\mathbf{W}_l \mathbf{l}_{t_{n+1}} + \mathbf{W}_d \mathbf{d}^u_{t_{n+1}} + \widetilde{\mathbf{b}}), \tag{4.9}$$

where the first term $\mathbf{W}_H \mathbf{H}^u_{t_n}$ denotes the user representation, and the second term $\mathbf{W}_l \mathbf{l}_{t_{n+1}} + \mathbf{W}_d \mathbf{d}^u_{t_{n+1}} + \widetilde{\mathbf{b}}$ denotes the spatial-aware location representation. In sum, the activity prediction task assists in the location prediction task, which in turn influences the activity representation learning during the model training with back-propagation, that is, they are interactively enhanced by each other. Furthermore, we consider two cases due to the presence of collective POIs: 1) if $y^u_{t_{n+1}} = 0$ meaning $l^u_{t_{n+1}}$ is an individual POI, we directly recommend individual POIs related to activity $c^u_{t_{n+1}}$; and 2) if $y^u_{t_{n+1}} = 1$, we further select individual POIs characterized by $c^u_{t_{n+1}}$ inside $l^u_{t_{n+1}}$ with higher ratings according to Observation 3.

4.3.4 Learning Strategy and Complexity Analysis

We adopt the Bayesian Personalized Ranking [98] to define the loss functions of activity and location prediction tasks. Followed by [34], we generate the training instances $\mathbf{\Omega}$ in a recursive way for each temporal-aware activity sequence and spatial-aware location sequence. For each instance, we then uniformly sample the negative activities and locations. In particular, regarding the collective POIs, the negative samples come from the nearby POIs as defined in Observation 2. The loss functions for the two prediction tasks are formulated by:

$$\begin{aligned}\mathcal{J}_c &= \sum\nolimits_{(c>c')\in\Omega} \ln\,(1+e^{-(r^u_{t,c}-r^u_{t,c'})}),\\ \mathcal{J}_l &= \sum\nolimits_{(l>l')\in\Omega} \ln\,(1+e^{-(r^u_{t,l}-r^u_{t,l'})}),\end{aligned} \tag{4.10}$$

where c' is the negative activity for c, and where l' is the negative location for l. Meanwhile, the loss function of POI type prediction (binary classification) is defined by:

$$\mathcal{J}_y = -\sum\nolimits_{k=1}^{|\Omega|} y_{t_k}\log(r^u_{t_k,y_{t_k}}) + (1-y_{t_k})\log(1-r^u_{t_k,y_{t_k}}). \tag{4.11}$$

Ultimately, we seek to minimize the sum loss:

$$\mathcal{J} = \lambda_c\mathcal{J}_c + \lambda_l\mathcal{J}_l + \lambda_y\mathcal{J}_y + \frac{\lambda}{2}||\Theta||^2, \tag{4.12}$$

where λ_c, λ_l, and λ_y ($\lambda_c + \lambda_l + \lambda_y = 1$) are weights to balance the importance of different losses; λ is the regularization coefficient; and $\mathbf{\Theta} = (\mathbf{W}, \mathbf{c}, \mathbf{y}, \mathbf{t}, \mathbf{l}, \mathbf{d}, \mathbf{b}, \widetilde{\mathbf{b}})$ is the set of model parameters to be learned. We use AdaGrad [99] to optimize the network parameters. Algorithm 2 shows the model training process composed of two parts: training instances generation (lines 2-4) and parameter update (lines 5-9).

Complexity Analysis. In the training process of iMTL, the computational complexity of the two-channel encoder and decoder is $O(D^2)$, where D is the embedding size. The complexity of recurrent units is $O(1)$. Consequently, given the training instances $\mathbf{\Omega}$ with the average sequence length $\bar{S}$, the overall complexity for each training iteration would be $O(|\Omega| \cdot \bar{S} \cdot D^2)$. In sum, the complexity of iMTL is linear to $|\Omega|$ and quadratic to the embedding size D.

4.4 Experiments and Analysis

We investigate the effectiveness of iMTL with the goal of answering the following three research questions.

RQ1: How does our proposed iMTL perform compared with the state-of-the-art methods w.r.t. both activity and POI recommendation tasks?

RQ2: How do different components of iMTL (e.g., fuzzy characterization strategy) affect its performance?

RQ3: How do hyper-parameters affect iMTL?

4.4.1 Experimental Setup

Datasets and Metrics. We conduct experiments on the three datasets as shown in Table 4.1. Following [34], we treat the first 80% sequences of each user as training set, the latter 10% as the validation set and the last 10% as test set. Note that for validation and test sets, we also remain the real visited individual POIs under collective POIs as ground-truth for POI prediction. Two standard metrics are adopted to evaluate the performance of all methods, namely Recall (Rec@K) and Mean Average Precision (MAP@K). Given a training/testing set with $|\Omega|$ samples, the two metrics are formulated as follows:

$$\text{Rec@}K = \frac{1}{|\Omega|}\sum_{i=1}^{|\Omega|}\frac{|S^i_{Rec}\cap S^i_{Visited}|}{|S^i_{Visited}|} \tag{4.13}$$

$$\text{MAP@}K = \frac{1}{|\Omega|}\sum_{i=1}^{|\Omega|}\frac{|S^i_{Rec}\cap S^i_{Visited}|}{\text{rank}_i} \tag{4.14}$$

where S^i_{Rec} is the top-K ranked list of recommended categories/POIs, and $S^i_{Visited}$ is the set of ground truth categories/POIs visited by users, rank_i denotes the position of the ground truth POI in the ranked list.

Comparison Methods. We compare iMTL with eight state-of-the-art approaches:

- **MostPop**: It recommends next activity and POI via popularity.
- **CateMF** [11]: It factorizes user-category and user-location matrices to generate the next activity and POI by considering the spatial factor and other similar users' preferences.
- **LBPR** [9]: It is a Listwise Bayesian Personalized Ranking method, which first recommends next activities and then predict next POIs by incorporating the geographical factor and the activity ranking influence.

- **ME** [4]: It is a state-of-the-art metric embedding model for next POI recommendation, which integrates the sequential and geographical context information and individual preference.

- **ST-RNN** [33]: It is an extended RNN method for next location recommendation that models local temporal and spatial contexts through the time-specific transition matrices and distance-specific transition matrices, respectively.

- **ATST-LSTM** [34]: It is an attention-based long and short-term memory network for next POI recommendation, which focuses on the relevant check-in behaviors in a trajectory by using spatiotemporal contextual information.

- **MCARNN** [7]: It is a multi-task context-aware recurrent neural network, which exploits the spatial-activity topic for next activity and location prediction.

- **HCT** [32]: It is a state-of-the-art approach for a next activity and POI recommendation with uncertain check-ins, by modeling the multi-granularity category transitions and category hierarchies.

Note that if the predicted activity happens at a collective POI, we need to further recommend next precise individual POIs inside the collective POI given the activity. As those individual POIs are cold start ones, we rank such POIs related to the predicted activity via their ratings based on Observation 3.

Parameter Settings. The parameters for all the baselines are tuned to achieve the best results or set as suggested by the original papers. For iMTL, the weight matrices, embeddings and hidden states are randomly initialized over uniform distribution. We apply a grid search in $[20, 200]$ stepped by 20 to find the optimal settings for embedding size $D_c = D_y = D_t = D_l = D_d$, and they are set to 120/140/120 for CLT, CAL and PHO, respectively (Fig. 4.5(a)). The number of recurrent layers is 3; the learning rate $\eta = 0.0001$; the iteration numbers are 25/20/30 for CLT, CAL and PHO, respectively (Fig. 4.5(b)); $\lambda_c = 0.4$, $\lambda_y = \lambda_l = 0.3$, $\lambda = 0.0025$ (Fig. 4.5(c)). The distance threshold $\triangle d = 2km$ for searching the negative POIs near the collective POI.

Table 4.2: Performance comparison of activity and location tasks on the three datasets. The best performance is boldfaced; the runner up is labeled with '*'; 'Improve' refers to the improvements (Paired t-test with p-value < 0.01) that iMTL achieves relative to the '*' results.

Activity Prediction Task						
	CLT		CAL		PHO	
	Rec@10	MAP@10	Rec@10	MAP@10	Rec@10	MAP@10
MostPop	0.1321	0.0435	0.1063	0.0417	0.1102	0.0430
CateMF	0.1516	0.0482	0.1238	0.0454	0.1214	0.0506
ME	-	-	-	-	-	-
LBPR	0.1829	0.0558	0.1425	0.0618	0.1647	0.0624
ST-RNN	-	-	-	-	-	-
ATST-LSTM	-	-	-	-	-	-
MCARNN	0.1935	0.0621	0.1579	0.0751	0.1974	0.0776
HCT	0.1984*	0.0614*	0.1602*	0.0764*	0.2010*	0.0781*
iMTL	**0.2213**	**0.0709**	**0.1724**	**0.0887**	**0.2311**	**0.0947**
Improve	11.5%	15.5%	7.6%	16.1%	15.0%	21.3%
Location (POI) Prediction Task						
	CLT		CAL		PHO	
	Rec@10	MAP@10	Rec@10	MAP@10	Rec@10	MAP@10
MostPop	0.0305	0.0104	0.0317	0.0119	0.0323	0.0125
CateMF	0.0323	0.0124	0.0341	0.0155	0.0352	0.0180
ME	0.0401	0.0137	0.0416	0.0205	0.0434	0.0207
LBPR	0.0446	0.0174	0.0483	0.0226	0.0525	0.0218
ST-RNN	0.0421	0.0162	0.0479	0.0230	0.0506	0.0243
ATST-LSTM	0.0465	0.0201	0.0522	0.0328	0.0591	0.0260
MCARNN	0.0458	0.0210	0.0545	0.0364	0.0608	0.0275
HCT	0.0477*	0.0204*	0.0617*	0.0372*	0.0662*	0.0301*
iMTL	**0.0534**	**0.0238**	**0.0691**	**0.0443**	**0.0769**	**0.0352**
Improve	11.9%	13.3%	12.0%	16.4%	16.2%	16.9%

4.4.2 Performance Comparison (RQ1)

Table 4.2 presents the performance (Rec@K and MAP@K) of all methods across the three datasets. Note that, ME, ST-RNN and ATST-LSTM do not consider category information, thus cannot predict next activity.

In terms of both tasks, the non-RNN based methods (Pop, CateMF and ME, LBPR) generally perform worse than RNN based baselines (ST-RNN, ATST-LSTM, MCARNN), demonstrating the efficacy of RNN on modeling the sequential dependency. As for RNN

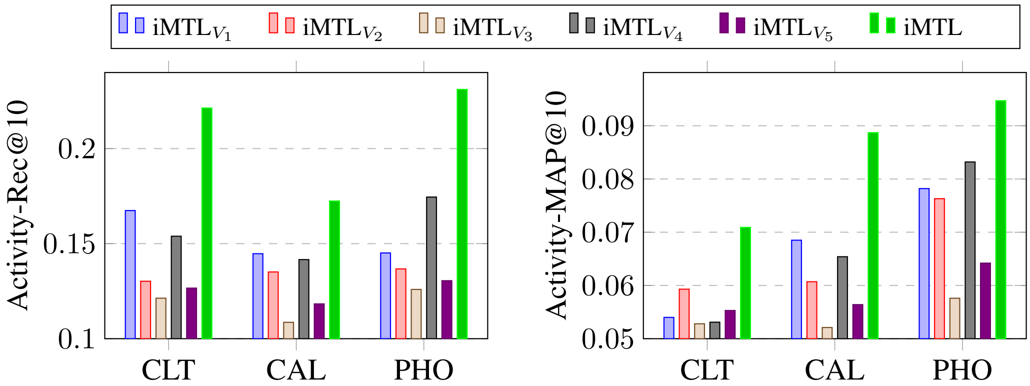


Fig. 4.3: Performance comparison for variants of iMTL on the three datasets w.r.t activity prediction task.

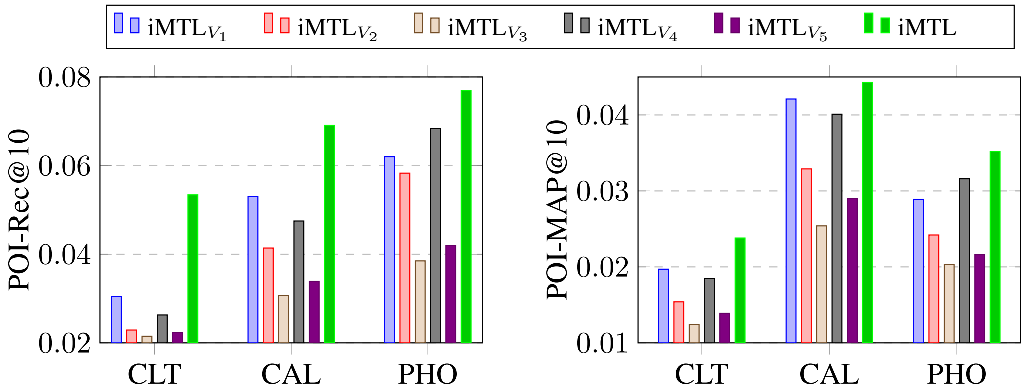


Fig. 4.4: Performance comparison for variants of iMTL on the three datasets w.r.t POI prediction task.

based methods, ATST-LSTM performs better than ST-RNN, which indicates the effectiveness of attention mechanism in modeling check-in sequence. They, however, are outperformed by MCARNN, due to its ability of capturing both sequential activities and locations via the MTL method. HCT performs better than RNN based methods, as it carefully designs and models the hierarchical dependencies between activity and location to ease the issue of next POI recommendation with uncertain check-ins. Overall, our proposed iMTL consistently achieves the best performance w.r.t. the two tasks across the three datasets. This is mainly because: (1) iMTL models user's uncertain activity via fuzzy characterization strategy; (2) iMTL adopts the auxiliary task (i.e., POI type prediction) to assist in activity prediction; and (3) iMTL exploits the interplay between activity and POI via an interactive multi-task learning framework.

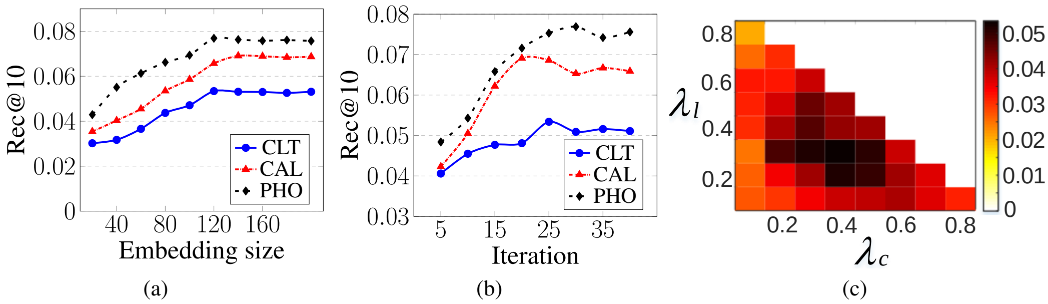


Fig. 4.5: Parameter sensitivity analysis on POI prediction task.

4.4.3 Detailed Study of iMTL (RQ2)

To investigate the effectiveness of different components of iMTL, we compare it with five different variants: (1) iMTL$_{w/o\ fuz}$ (iMTL$_{V_1}$) removes the fuzzy characterization on users' uncertain activities. Following [32], we directly use the category of a collective POI to represent the activity, instead of the fuzzy representation; (2) iMTL$_{w/o\ aux}$ (iMTL$_{V_2}$) removes the auxiliary task, i.e., POI type prediction; (3) iMTL$_{w/o\ fuz-aux}$ (iMTL$_{V_3}$) ignores both fuzzy characterization and auxiliary task; (4) iMTL$_{w/o\ 2c}$ (iMTL$_{V_4}$) merges the two-channel encoder into one by using multi-modal embedding manner; and (5) iMTL$_{w/o\ inter}$ (iMTL$_{V_5}$) removes the interactive learning method by performing the three prediction tasks in parallel.

We report the results in Fig. 4.3 and Fig. 4.4, where iMTL significantly outperforms its variants regarding to both activity and location prediction tasks. We notice that iMTL$_{w/o\ fuz-aux}$ performs worse than either iMTL$_{w/o\ fuz}$ or iMTL$_{w/o\ aux}$, which suggests that both fuzzy characterization strategy and auxiliary task indeed improve the recommendation performance. Generally, the performance decrease of iMTL$_{w/o\ aux}$ far exceeds that of iMTL$_{w/o\ fuz}$, implying that the auxiliary task (i.e., POI type prediction) plays a more important role than the fuzzy characterization strategy. Both iMTL$_{w/o\ 2c}$ and iMTL$_{w/o\ inter}$ underperform iMTL, indicating the advantages of the two-channel encoder and interactive multi-task learning strategy. In summary, our proposed iMTL benefits from the four delicately designed components.

4.4.4 Parameter Sensitivity Analysis (RQ3)

Fig. 4.5 depicts the results (Rec@10) of parameter sensitivity analysis on POI prediction and similar trends can be observed with other settings of K. In Fig. 4.5(a), the performance of iMTL climbs up as the embedding size increases, and gradually becomes stable with the size around 120. We further study the convergence property of iMTL as shown in Fig. 4.5(b), where we observe that iMTL can converge within 30 iterations on the three datasets. Fig. 4.5(c) shows the performance of varying the combination weights λ_c and λ_l, which control the importance of the activity and location prediction tasks, respectively. Note that the importance of the POI type prediction task λ_y is determined by $1-\lambda_c-\lambda_l$. From the results, we find that iMTL performs best with $\lambda_c = 0.4$, $\lambda_l = \lambda_y = 0.3$, which implies that all the three tasks are vital for more accurate next POI recommendation with uncertain check-ins.

4.5 Summary

In this chapter, we propose an interactive multi-task learning (iMTL) framework for the next POI recommendation with uncertain check-ins. In particular, we devise a two-channel encoder, i.e., temporal-aware activity encoder and spatial-aware location preference encoder, to capture the transitions of activities and locations, whereby a fuzzy characterization strategy is proposed to better represent activity over uncertain check-ins. The task-specific decoder then interactively aggregates the latent representations of the two-channel encoder to perform both activity and location prediction tasks. Experimental results show the superiority of iMTL over the state of the arts w.r.t next activity and location recommendation tasks.

Chapter 5

CART: Conversation-based Adaptive Relational Translation Method[1]

In Chapter 3 and 4, we focus on exploiting context features from users' historical trajectories into representation learning based model or multi-task learning based model to help resolve the issue of uncertain check-ins in the next POI recommendation task. However, existing next POI recommendation methods (i.e., static methods), including HCT and iMTL, are hard to capture user's immediate preference and incorporate user's feedback regarding the recommendation results. Conversational recommendation systems (CRS) bring promising potential in allowing the system to acquire user's immediate preference via interactions and adapt to user's feedback. We argue that the value of a CRS for the next POI recommendation task has not reached its full potential, since an efficient location-aware CRS should delicately accommodate both the rich context of users' check-in behaviors and the merit of multi-round CRS.

In this chapter, we propose a conversation-based adaptive relational translation method (CART) for next POI recommendation over uncertain check-ins. It is equipped with recommender and conversation modules to interactively acquire users' immediate preference and make dynamic recommendations. Specifically, the recommender built upon the adaptive relational translation method performs location prediction via modeling both users' historical sequential behaviors and the immediate preference received from conversations;

[1]The work in this chapter is accepted by IEEE Transactions on Neural Networks and Learning Systems [100], 2022.

and the conversation module aims to achieve successful recommendations in fewer conversation turns by learning a conversational strategy, whereby the recommender can be updated via the user response. Extensive experiments on four real-world datasets show the superiority of our proposed CART over the state-of-the-art methods.

The remaining of this chapter is organized as follows: In Section 5.1, we first propose the CART framework, which is composed of recommender and conversation modules. Section 5.2 presents the experimental results, followed by the conclusion in Section 5.3.

5.1 The CART Framework

Fig. 5.1 depicts the architecture of our proposed CART under the multi-round setting, aiming to simulate the conversation process, that is, enquiring a user's preference towards attributes or making recommendations until the user accepts the recommendation or chooses to quit. This enables users to interact with an agent via iterative conversations and then performs recommendation by two modules: (1) **Recommender Module**, which is an adaptive relational translation method to jointly model a user's historical sequential behaviors and the immediate preference obtained in conversations; and (2) **Conversation Module**[2], which seeks the best strategy for action selection and achieves a successful recommendation with the fewer conversation turns. In particular, it consists of a *State Vector*, *Policy Network* and *User Response*. At each conversation turn, the *State Vector* serves as a bridge between the recommender and conversation modules, encoding the user preference and conversation history. The *Policy Network* plays a role of action generator, which decides on whether to ask an attribute (e.g., a category) or make a recommendation (e.g., a specific POI) based on the input state vector, and returns an up-to-date reward according to the user response.

Notations. Let $u \in \mathcal{U}$ denote a user u from the user set $\mathcal{U}$ and $l \in \mathcal{L}$ denote a POI (i.e., location) from the POI set $\mathcal{L}$. For each user u, we order all his check-ins by timestamp, and then split them into sequences by day, denoted as $\mathcal{S}^u$. Thus, the j-th check-in sequence of u refers to a set of time-ordered check-ins within a day: $\mathcal{S}^{u,j}$ =

[2]Following the user simulation settings in [30, 31], we use templates as wrappers to interact with a user for conversation process rather than considering language understanding and generation.

Table 5.1: Notations.

Notations	Descriptions
$\mathcal{U}, \mathcal{L}, \mathcal{C}$	User set, location set, category set
u, l, c, t	User $u \in \mathcal{U}$, location $l \in \mathcal{L}$, category $c \in \mathcal{C}$, time t
g, y	Geographical region, POI type
$\mathcal{S}^u$	Check sequences of user u
$\mathcal{S}^{u,j}$	The j-th check sequence of user u: $\mathcal{S}^{u,j} \in \mathcal{S}^u$
$s^u_{t_i}$	Check-in activity of u at t_i: $s^u_{t_i} = (l_{t_i}, t_i, c_{t_i}, g_{t_i}, y_{t_i})$
$\mathbf{u}, \mathbf{l}, \mathbf{t}, \mathbf{c}, \mathbf{g}, \mathbf{y}$	Embeddings associated with u, l, t, c, g, y
$\mathcal{P}_u$	u's immediate preferences obtained in conversations
α, β	Attention weights of relations
$\mathbf{v}$	State vector
$\mathbf{r}^{seq}, \mathbf{r}^{imm}$	Relation embeddings
$d(u, l)$	Distance between u and l
P_{u,l_t}	Probability of u visiting l at time t
$\mathcal{A}$	Action space
Re^+_*, Re^-_*	Positive/negative reward
$\pi_\theta(a_n\|\mathbf{v}_n)$	Policy network and $a_n \in \mathcal{A}$
γ	Discount factor

$\{s^u_{t_1}, s^u_{t_2}, \ldots, s^u_{t_n}\}$, $\mathcal{S}^{u,j} \in \mathcal{S}^u$; each check-in $s^u_{t_i} = (l_{t_i}, t_i, c_{t_i}, g_{t_i}, y_{t_i})$ means that a user u visits a POI l_{t_i} at geographical region g_{t_i} and time t_i (timestamps are discretized into 24 slots in a day); $c_{t_i} \in \mathcal{C}$ is the category of l_{t_i}; and y_{t_i} is the type of l_{t_i}, where $y_{t_i} = 0$ refers to an individual POI, otherwise a collective POI. We list some important notations in Table 5.1.

Research Problem. Given user u's historical check-in sequence $\mathcal{S}^{u,j} = \{s^u_{t_1}, s^u_{t_2}, \ldots, s^u_{t_k}\}$ and the next check-in time t_{k+1}, our goal is to recommend the next POI $l_{t_{k+1}}$ in multi-round conversations by maximizing the accumulated reward given the limited conversation turns. Note that if the recommended $l_{t_{k+1}}$ is a collective POI, we further predict individual POIs inside $l_{t_{k+1}}$.

5.1.1 Recommender Module

With the aim of making more accurate next POI recommendations over uncertain check-ins, we design a translation-based recommender under the multi-round setting of CRS, which is capable of modeling the rich context information (e.g., spatiotemporal context and immediate preference) of location visits. Hence, the proposed recommender not

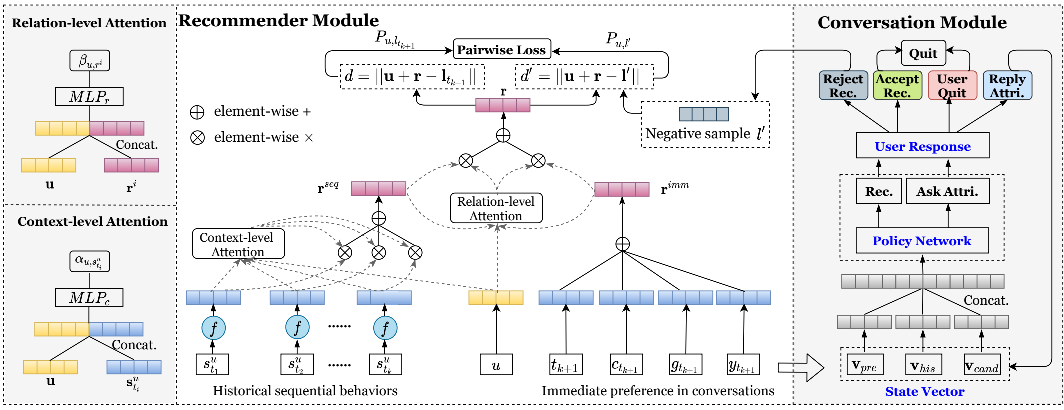


Fig. 5.1: The framework of CART, which is composed of recommender and conversation modules.

only delicately captures users' context-aware preference but also overcomes the inherent limitation of inner product in FM [40], which is adopted as the recommender in state-of-the-art CRS methods [25, 30].

5.1.1.1 Basic Translation Method

TransE is a representative method among the various knowledge graph based techniques [101, 102], which aims to embed the triples (h, r, t) into a transition space that satisfies $\mathbf{h} + \mathbf{r} \approx \mathbf{t}$, where h, r, t represent head entity, relation and tail entity, respectively[3]. It has been widely studied in recommender systems due to its promising ability over factorization-based methods [102, 103]. Hence, the affinity of user u and POI l is defined as:

$$d(u, l) = ||\mathbf{u} + \mathbf{r} - \mathbf{l}||_2^2. \tag{5.1}$$

Note that there are various translation-based methods [104–106] which can model different relational patterns, e.g., symmetric/asymmetric, inversion and composition. In our study, we mainly focus on taking advantage of translation-based methods to alleviate the inherent issue of factorization-based methods, instead of exploring the efficiency of different translation-based methods. We thus propose a simple yet effective TransE-based recommender. The key point is, therefore, to capture the translation vector $\mathbf{r}$ (i.e., the

[3]We use lowercase in bold (e.g., $\mathbf{h}, \mathbf{r}, \mathbf{t}, \mathbf{u}, \mathbf{l}$) to denote the embedding of the corresponding notation (e.g., h, r, t, u, l).

relation embedding encodes a user's spatiotemporal sequential check-in behaviors and immediate preference in conversations) in the following model description.

5.1.1.2 Adaptive Relational Translation Method

In the next POI recommendation scenario, the sequential regularity of a user's location visit is of significance in capturing his personalized preference [41, 56, 59]. Enlightened by the recent success of context-aware relation representation in [40], we construct an adaptive relation vector to translate a user u to the next POI l by considering both historical sequential behaviors $\mathcal{S}^{u,j} = \{s^u_{t_1}, s^u_{t_2}, \ldots, s^u_{t_k}\}$ and immediate preference $\mathcal{P}_u$ obtained in conversations, i.e., $(u, < \mathcal{S}^{u,j}, \mathcal{P}_u >, l)$, via attention-based mechanisms.

Since both $\mathcal{S}^{u,j}$ and $\mathcal{P}_u$ affect user u's next movement, they can be naturally treated as translation relations inspired by [40, 102]. We thus encode the two types of relations derived from the above two factors. Regarding $\mathcal{S}^{u,j}$, we adopt the context-level attention to capture u's personalized varying attentions on different historical check-ins:

$$\mathbf{s}^u_{t_i} = f(\mathbf{l}_{t_i}, \mathbf{t}_i, \mathbf{c}_{t_i}, \mathbf{g}_{t_i}, \mathbf{y}_{t_i}),\ s^u_{t_i} \in \mathcal{S}^{u,j} \tag{5.2}$$

$$o^s_{t_i} = \mathrm{MLP}_c([\mathbf{u}; \mathbf{s}^u_{t_i}]), \tag{5.3}$$

$$\alpha_{u,s^u_{t_i}} = \mathrm{softmax}(o^s_{t_i}) = \frac{\exp(o^s_{t_i})}{\sum_{i=1}^{k} \exp(o^s_{t_i})}, \tag{5.4}$$

where $\mathbf{s}^u_{t_i}$ encodes the context-aware check-in activity[4]; $f(\cdot)$ is the element-wise summation; $\mathrm{MLP}_c(\cdot)$ is a two-layer attention network (see bottom left of Fig. 5.1), and its input is the concatenation of $\mathbf{u}$ and $\mathbf{s}^u_{t_i}$; the softmax function is used to calculate the normalized impact weight; and k is the length of $\mathcal{S}^{u,j}$. Therefore, the sequential regularity can be encoded by the relation embedding $\mathbf{r}^{seq}$, which is the weighted sum of the check-in activity embeddings:

$$\mathbf{r}^{seq} = \sum\nolimits_{i=1}^{k} \alpha_{u,s^u_{t_i}} \cdot \mathbf{s}^u_{t_i}. \tag{5.5}$$

In our conversation scenario, we can acquire u's immediate preferred attributes $\mathcal{P}_u$ for next location visit at time step t_{k+1}, for example, Bob specifies a *Thai restaurant* (i.e., category) as his preferred attribute. A good relation needs to encode such acquired

[4] $\mathbf{c}_{t_i}$ is represented by the weighted combination of embeddings of all categories inside l_{t_i}, if $y_{t_i} = 1$ as in [35].

attributes as well as the time information to better localize his desired POIs, since the time factor is also a vital context information in users' next movement prediction [35,40], e.g., the check-ins of shops usually happen between 10am and 6pm; while the check-ins of nightlife spots are more likely to occur after 7pm. We thus generate the relation embedding $\mathbf{r}^{imm}$ based on $\mathcal{P}_u$:

$$\mathbf{r}^{imm} = \sum\nolimits_{p_i \in \mathcal{P}_u} \mathbf{p}_i + \mathbf{t}_{k+1}, \tag{5.6}$$

where p_i would be one of the preferred attributes at t_{k+1} in the conversation session, e.g., $c_{t_{k+1}}$, $g_{t_{k+1}}$, $y_{t_{k+1}}$.

Although the above two relations (i.e., r^{seq} and r^{imm}) contribute to the next preferred POI prediction, different users are supposed to have varying attention to different relations due to users' complex behavioral preference. To capture the importance of such two relations, we further propose a relation-level attention to calculate the impact weight:

$$o_{r^i} = \mathrm{MLP}_r([\mathbf{u}; \mathbf{r}^i]), \tag{5.7}$$

$$\beta_{u,r^i} = \mathrm{softmax}(o_{r^i}) = \frac{\exp(o_{r^i})}{\sum_{i=1}^{|\mathcal{R}|} \exp(o_{r^i})}, \tag{5.8}$$

where $\mathrm{MLP}_r(\cdot)$ is a two-layer attention network (see upper left of Fig. 5.1), and its input is the concatenation of $\mathbf{u}$ and $\mathbf{r}^i$; $|\mathcal{R}|$ is the number of relations, and two types of relations r^{seq} and r^{imm} are considered in our study. As a result, the final relation embedding $\mathbf{r}$ in Eq.(5.1) is formulated as,

$$\mathbf{r} = \sum\nolimits_{i=1}^{|\mathcal{R}|} \beta_{u,r^i}\, \mathbf{r}^i. \tag{5.9}$$

Finally, given u's historical sequential behaviors $\mathcal{S}^{u,j}$ and immediate preference $\mathcal{P}_u$ in the conversation session, the probability of u visiting $l_{t_{k+1}}$ at t_{k+1} is predicted by:

$$P_{u,l_{t_{k+1}}} \propto \frac{1}{d(u, l_{t_{k+1}})}. \tag{5.10}$$

5.1.2 Conversation Module

The conversation module consists of a *State Vector*, a *Policy Network* and a *User Response*. The *State Vector* serves as the bridge between the recommender and conversation modules; the *Policy Network* aims to learn the strategy of action selection to determine whether to ask an attribute or make a recommendation; and the *User Response* returns user feedback on such an action selection, whereby the immediate user preference can be obtained.

5.1.2.1 State Vector

The state vector encodes a user's preference over different attributes and the conversation history. As such, the state vector $\mathbf{v}$ is represented by the concatenation of three components:

$$\mathbf{v} = [\mathbf{v}_{pre}; \mathbf{v}_{his}; \mathbf{v}_{cand}], \tag{5.11}$$

$$\tilde{d}(u, p) = ||(\mathbf{u} + \sum\nolimits_{p_i \in \mathcal{P}_u} \mathbf{p}_i + \mathbf{t}_{k+1}) - \mathbf{p}||_2^2, \tag{5.12}$$

where each dimension of $\mathbf{v}_{pre} \in \mathbb{R}^{|\mathcal{P}|}$ encodes the user preference on each attribute p, evaluated by Eq.(5.12); $|\mathcal{P}|$ is the size of all attributes[5]. The term $(\mathbf{u} + \sum_{p_i \in \mathcal{P}_u} \mathbf{p}_i + \mathbf{t}_{k+1})$ represents the context-aware user representation by considering user u's immediate preference $\mathcal{P}_u$ at time step t_{k+1}. $\mathbf{v}_{his} \in \mathbb{R}^N$ (N is the maximum conversation turn) encodes the conversation history, where each dimension encodes u's feedback (i.e., reward) at each turn n, and we will detail the settings on reward in the *Policy Network*. $\mathbf{v}_{cand}$ encodes the size of current candidate POI set ($|\mathcal{L}_{cand}|$) via the binary features [30], where $|\mathcal{L}_{cand}|$ will be reduced with the increasing of preferred attributes in $\mathcal{P}_u$, and the recommendation action will be triggered if $|\mathcal{L}_{cand}|$ is smaller than a threshold (L_{min}) to avoid asking more attributes.

5.1.2.2 Policy Network and User Response

The action selection is derived from the policy network in the conversation module. Inspired by [25,30], we adopt a two-layer neural network which can be optimized with the standard policy gradient method. Specifically, the input is generated by the *State Vector*, and the output is normalized to be a probability distribution over all actions by a softmax operation. Our action space consists of all attributes of POIs and a recommendation action, i.e., $\mathcal{A}^{|\mathcal{P}|+1} = \{\{a_{ask}(p), p \in \mathcal{P}\} \cup a_{rec}\}$. At each turn, the conversation module

[5] The attributes considered in this study include categories, geographical regions and types of POIs.

will take an action determined by the *Policy Network* and receive an up-to-date reward from the *User Response* (i.e., a user provides positive/negative response on either a_{ask} or a_{rec}). We define the rewards as follows:

- Re^+_{ask} / Re^-_{ask}: a positive/negative reward when the user gives positive/negative feedback on the asked attribute;

- Re^+_{rec} / Re^-_{rec}: a positive/negative reward when the user accepts/rejects the recommended POIs;

- Re^-_{quit}: a strongly negative reward if the user quits the conversation, i.e., reaching the maximum turn N.

In the scenario of uncertain check-ins, the proposed CRS interacts with a user to obtain his immediate preference, e.g., the desired activity and POI type, which helps to locate the desired individual POI inside a collective POI. We thus specially design an auxiliary reward w.r.t. Re^+_{rec} inspired by the observations in [35]. That is, a user's check-in behaviors at a collective POI l are affected by the diversity of l, since users usually prefer collective POIs with more choices. Such an auxiliary reward is considered when the user accepts the recommended collective POI l:

$$Re^+_{rec} = \begin{cases} Re^+_{rec}, & y = 0 \\ Re^+_{rec} + \omega \frac{m_p}{M_l}, & y = 1 \end{cases} \tag{5.13}$$

where $\frac{m_p}{M_l}$ is the auxiliary reward; m_p is the number of individual POIs w.r.t. the preferred attribute p, e.g., two individual POIs (l_4 and l_5) belong to Thai restaurant in Fig. 1.3(b); and M_l is the total number of individual POIs in l, e.g., $M_{l_3} = 8$, as l_3 contains eight individual POIs; ω is the normalization coefficient.

We denote the policy network as π_θ and $\pi_\theta(a_n|\mathbf{v}_n)$ represents the probability of taking an action $a_n \in \mathcal{A}$ given the state $\mathbf{v}_n$ at the n-th turn. Following [107], the policy network is optimized through reinforcement learning (RL):

$$\theta \leftarrow \theta + \eta \bigtriangledown \log \pi_\theta(a_n|\mathbf{v}_n)\bar{Re}_n, \ \theta \in \Theta \tag{5.14}$$

where $\theta \in \Theta$ is the parameter of the policy network; η is the learning rate; $\bar{Re}_n = \sum_{n'=n}^{N} \gamma^{n'} Re_{n'}$ is the sum of rewards from turn n to the final turn N; and γ is a discount factor.

5.1.3 Connection of Recommender and Conversation Modules

Once a recommendation action is triggered in the conversation module, the recommender needs to suggest top-K POIs for user u. To this end, it first constructs the relation vector $\mathbf{r}$ derived from user u's historical check-ins $\mathcal{S}^{u,j}$ and immediate preference $\mathcal{P}_u$ in conversations using Eqs.(5.5-5.9), and then calculates the probability of user u visiting each candidate POI based on Eq.(5.10). Note that the rejected POIs in conversations will be treated as negative samples to update the recommender. In the scenario of uncertain check-ins, considering user u's desired POI could be a collective POI l we propose a rating-based sampling strategy, i.e., $l_i \sim rate(l_i)/\sum_{i=1}^{M_l} rate(l_i)$, to select the individual POI $l_{t_{k+1}}$ corresponding to user u's preferred attributes $\mathcal{P}_u$ inside POI l, where $rate(l_i)$ is the rating of POI l_i. This helps achieve a relatively fair recommendation rather than simply selecting such individual POIs with higher ratings as in [35].

5.1.4 Discussion on the Issue of Uncertain Check-ins

We highlight the overview of CART in dealing with the issue of uncertain check-ins in next POI recommendation from three aspects. **(1)** The uncertain check-ins over collective POIs increase the difficulty of predicting accurate user activity (i.e., category) of next movement. The proposed CRS enables the system to interact with users to acquire, for example, their desired activities and POI types, which is one of the key challenges for a successful next POI recommendation under the uncertain check-in scenario. **(2)** In the scenario of uncertain check-ins, a user's check-in behaviors at a collective POI l are affected by the diversity of l, since users usually prefer collective POIs with more choices [35]. We thus specially design an auxiliary reward when the user accepts the recommended collective POI l. **(3)** Considering a user's desired POI l could be a collective POI, we need to further select an individual POI inside l. Hence, we design a rating-based sampling strategy for fairly choosing individual POI corresponding to the user's activity inside a collective POI, rather than simply selecting individual POIs with higher ratings.

5.1.5 Model Optimization

Most existing static next POI recommendation studies directly treat the non-interacted POIs as negative samples to optimize the Bayesian Personalized Ranking (BPR) or Binary Cross-Entropy (BCE) objectives [9,33,35,41,46], assuming that the interacted POIs should gain higher prediction scores than those not being interacted with. However, users do not explicitly indicate that they dislike those non-interacted POIs, which results in the lack of true negative samples for such static models. In CRS, we are not only able to acquire a user's immediate preference but also explicitly obtain rejected POIs from his response. We thus take such rejected POIs as the negative samples to train the recommender. The loss function is defined via a BPR loss,

$$\mathcal{J} = -\sum_{u \in \mathcal{U}} \sum_{\mathcal{S}^{u,j} \in \mathcal{S}^u} \sum_{l' \in \mathcal{L}^u_{rej}} \ln \sigma(P_{u,l_{t_{k+1}}} - P_{u,l'}) + \lambda ||\bar{\Theta}||^2, \tag{5.15}$$

where $\mathcal{L}^u_{rej}$ is a set of rejected POIs for u received from conversations; λ is the regularization coefficient; and $\bar{\Theta}$ is the parameter set of the recommender.

Note that, we do not train our proposed CART from scratch. Following [30, 31], our training process comprises two stages: (1) the offline training aims to pre-train the recommender and conversation modules on the training set, where the maximum entropy strategy is used for determining which attribute to ask; and (2) in the online training stage, the pre-trained CRS is optimized by interacting with the user simulator through the conversation module on the training set. Algorithm 3 shows the online training of the CART. In particular, at each conversation turn n, the conversation module performs state generation and action selection (lines 6-7). If an action a_n asking an attribute p (i.e., $a_{ask}(p)$) is accepted by u, p will be added into u's preferred attribute set $\mathcal{P}_u$; otherwise $a_{ask}(p)$ will be removed from u's action space $\mathcal{A}_u$ (lines 8-12). If a_n is making a recommendation (i.e., a_{rec}), then the top-K POI list $\mathcal{L}^u_{rec}$ is generated by the recommender. The conversation succeeds & quits if $\mathcal{L}^u_{rec}$ contains u's desired POI; otherwise such rejected POIs are removed from the POI candidate set $\mathcal{L}^u_{cand}$ but added into $\mathcal{L}^u_{rej}$ (explicit negative samples) for updating the recommender (lines 13-20). Finally the policy network gets updated if the conversation session succeeds or quits (line 21). Moreover, once the model has been well-trained, CART performs incremental training

Algorithm 3: The online training for CART

Input: $\mathcal{U}, \mathcal{L}, \mathcal{S}, N, \mathcal{P}, L_{min}, \omega, \eta, \gamma, \lambda$

Output: Top-K recommendation list

Initialize model parameters Θ and $\bar{\Theta}$;

foreach $u \in \mathcal{U}$ **do**

  foreach *turn* $n = 1, 2, \ldots, N$ **do**

    if $n > N$ **then**

      Fail & quit

    `// State Vector`

    Compute state $\mathbf{v}_n$ according to Eq.(5.11)

    `// Policy Network`

    Take an action $a_n \sim \pi_\theta(\cdot|\mathbf{v}_n)$

    `// User Response`

    if $a_n = a_{ask}(p)$ **then**

      if u *accepts* p **then**

        $\mathcal{P}_u = \mathcal{P}_u \cup p$, get a reward Re^{+}_{ask}

      else

        $\mathcal{A}_u = \mathcal{A}_u \setminus a_{ask}(p)$, get a reward Re^{-}_{ask}

    else

      Recommend top-K POIs $\mathcal{L}^u_{rec} \subset \mathcal{L}^u_{cand}$

      if u *accepts* $\mathcal{L}^u_{rec}$ **then**

        Succeed & quit, get a reward Re^{+}_{rec}

      else

        $\mathcal{L}^u_{cand} = \mathcal{L}^u_{cand} \setminus \mathcal{L}^u_{rej}$

        $\mathcal{L}^u_{rej} = \mathcal{L}^u_{rec}$, get a reward Re^{-}_{rec}

        Update recommender $\bar{\Theta}$ using $\mathcal{L}^u_{rej}$

  Update policy network Θ

with the ever-growing of users' check-in records, instead of updating it from scratch. In particular, if there are new check-in records for some users coming, we will only update the relevant model parameters (e.g., user and POI embeddings) for these users.

5.2 Experiments and Analysis

To evaluate the performance of our proposed CART, we conduct experiments on four real-world datasets with the goal of answering the following research questions.

Table 5.2: Statistics of the four datasets.

Dataset	#User	#POI	#1st-Layer	#2nd-Layer	#Check-in
CAL	301	985	9	184	13,954
CHA	1,580	1,791	10	239	20,940
PHO	1,623	2,441	8	251	22,620
SIN	2,676	3,440	15	264	116,757

RQ1: does our proposed CART outperform the representative CRS methods?

RQ2: how does our CART compared with state-of-the-art static recommendation methods?

RQ3: how does the quality of user response affect CART?

RQ4: how do different components of CART affect its performance?

RQ5: how do different hyper-parameters affect CART?

5.2.1 Experimental Setup

5.2.1.1 Datasets

We perform experiments on four real-world datasets utilized in [32, 35] that contain collective POIs constructed from Foursquare in four cities, i.e., Calgary (CAL), Charlotte (CHA), Phoenix (PHO) and Singapore (SIN)[6], as summarized in Table 5.2. Following [30, 102], we remove the users with less than 10 interactions and split the sequences of each user in the ratio of 7:2:1 for training, validation and test.

5.2.1.2 Action Space

In real application, asking an attribute from a large attribute space leads to lengthy conversations. Following [30], we consider the 1st-layer category provided by *Foursquare Venue Category Hierarchy*[7] as the parent attributes, e.g., 'food' is the 1st-layer category which contains several 2nd-layer child categories {'Korean restaurant', 'Malay restaurant', $\cdots$, 'Thai restaurant'}. As depicted by Table 5.2, the number of 1st-layer category

[6]Note that there is no rating information in SIN dataset, we thus use the popularity (i.e., the check-in frequency of a POI) as a substitute since it is a significant factor in recommendation [108]. Specifically, we evenly divided such frequency into five groups corresponding to the range of ratings [1,5].

[7]https://developer.foursquare.com/docs/build-with-foursquare/categories/

is far less than that of the 2nd-layer category. In our location-service scenario, the parent attribute of geographical region contains child attributes such as $\{g_1, g_2, g_3, g_4\}$ in Fig. 1.3(b). Similarly, the parent attribute of POI type contains such child attributes $y = \{0, 1\}$. As a result, we obtain the shrinking attribute space, which only contains the parent attributes of category, geographical region and POI type. Taking CAL as an example, its final output (i.e., action space) size of the policy network is 12, that is, 9 (1st-layer category)+1 (geographical region)+1 (POI type)+1 (recommendation) = 12. Following [30], the system selects a parent attribute to ask, and the user can reply with multiple child attributes.

5.2.1.3 User Simulator

The CRS needs to interact with users to make recommendations, which is expensive to acquire real dialogue resources [29]. Following [25, 30], we create a user simulator to enable CART to be trained and evaluated in the interactive process. Given u's historical sequential behaviors $\mathcal{S}^{u,j}$ (e.g., $l_1 \to l_2$), the user simulator aims to simulate a conversation session for an observed interaction (u, l), where l is treated as u's desired POI to seek for next movement and $\mathcal{P}_l$ is the oracle set of attributes preferred by u in this session. As such, the session is initialized by a random selected attribute from $\mathcal{P}_l$ and then goes in the loop as shown in Algorithm 3.

5.2.1.4 Evaluation Metrics

We adopt two standard metrics to evaluate the CRS methods by following [30]: (1) success rate (SR@n), which measures the ratio of successful conversations, i.e., correctly recommending the desired POI by turn n; and (2) average turns (AT), which records the average turns (i.e. average conversation length) needed to end the session. As such, higher SR denotes better recommendation, and smaller AT represents more efficient conversations. Specifically, SR@n and AT are defined by:

$$SR@n = \frac{\sum_{i=1}^{n} \#\text{successful\_conversations}(i)}{\#\text{conversations}}, \tag{5.16}$$

$$AT = \frac{\sum_{i=1}^{n} i \times \#\text{successful\_conversations}(i)}{\sum_{i=1}^{n} \#\text{successful\_conversations}(i)}, \tag{5.17}$$

where #successful_conversations(i) is the number of successful conversations (i.e., successful test samples) at turn i among all the test samples; #conversations is equal to the number of all test samples. Additionally, to compare the performance of static state-of-the-arts with CART, we adopt recall (Recall@K) and mean reciprocal rank (MRR@K) to measure the ranking quality for the top-K recommended POIs. Note that we calculate the recall and MRR of top-K POIs by turn n for the CART, that is, the recall and MRR are the accumulated scores from conversation turn 1 to turn n. Note that we calculate the recall and MRR for the CART after the conversation finishes.

5.2.1.5 Baselines

Following [30], we compare the proposed CART with five state-of-the-art CRS methods:

- **Max Entropy**: It is a rule-based method which aims to select an attribute with the maximum entropy within the current candidate POIs, and such system follows certain probability to make recommendations as in [30].

- **CRM** [25]: It is a single-round CRS method which consists of a belief tracker to encode the state vector and the policy network to perform action selection. We follow [30] to adapt it to a multi-round setting without considering natural language understanding module; therefore the adapted CRM serves as an upper bound study.

- **Qrec** [28]: It is a question-based recommender system based on the extended matrix factorization, which is originally designed for interactively asking questions based on the descriptions and reviews of items. To generate the question pool, we use the item attributes in this work.

- **EAR** [30]: It is a state-of-the-art approach with a multi-round CRS setting, where a three-stage solution is devised and its recommender is built upon factorization machine [36].

- **SCPR** [31]: it is a CRS method that performs interactive path reasoning on a graph under the multi-round setting.

The recent CRS methods, such as [26,79,82], are beyond the scope of our study, since they either let the CRS recommend items without asking users' preferred attributes or focus on natural language understanding and generation. Besides, we compare with nine state-of-the-art static methods for next POI recommendation:

- **MostPop**: It recommends the next POI via popularity.
- **LBPR** [9]: It is a listwise Bayesian personalized ranking method, which predicts the next category and POI in a two-fold way.
- **ST-RNN** [33]: It is a RNN-based method which models temporal and spatial context with time- and distance-specific transition matrices.
- **MCARNN** [7]: It is a multi-task learning framework to capture a user's activity and location preferences.
- **STA** [40]: It is a translation-based method to model the users' spatiotemporal context;
- **PLSPL** [56]: It is a unified framework to characterize a user's long-term and short-term preferences.
- **HCT** [32]: It is the pioneer work to handle the next POI recommendation with uncertain check-ins.
- **iMTL** [35]: It is a multi-task learning framework to jointly model a user's context of check-in behaviors and targets at resolving the issue of uncertain check-ins.
- **CART-CRS**: It is a variant of our CART, which removes the conversion module and only encodes a user's historical sequential behaviors (see 'Recommender module' in Fig. 5.1) for next POI recommendation.

5.2.1.6 Training Details and Parameter Settings

We tune all hyper-parameters on the validation set and empirically find out the optimal parameter settings for all the methods. Specifically, the embedding size is searched in the range of $[20, 200]$ stepped by 10; the learning rate and regularization coefficient are searched in $\{0.0001, 0.0005, 0.001, 0.005, 0.01, 0.05, 0.1\}$. For our proposed CART, the size of embeddings is 50, which are randomly initialized over uniform distribution; we adopt SGD optimizer with learning rate and regularization coefficient being 0.001; the rewards are empirically set as: $Re^{+}_{ask} = 1$, $Re^{-}_{ask} = -1$, $Re^{+}_{rec} = 2$, $Re^{-}_{rec} = -2$ $Re^{-}_{quit} = -3$; the discount factor $\gamma = 0.7$; the normalization coefficient $\omega = 2$; the max turn $N = 10$; the threshold to trigger recommendation action $L_{min} = 10$; the length of the recommendation list $K = 10$; and the number of recommended individual POI inside a collective POI $K_1 = 3$. For the settings of CRS baselines (i.e., CRM, Qrec, EAR and SCPR), the embedding size is $\{60, 100, 60, 60\}$; the learning rate and regularization coefficient are $\{0.001, 0.1, 0.01, 0.01\}$ and $\{0.01, 0.01, 0.001, 0.001\}$, respectively; other parameters are set as suggested in the related papers. Regarding the settings of static baselines, the embedding size and the learning rate for LBPR, ST-RNN, MCARNN, STA, PLSPL, HCT and iMTL are set to $\{100, 100, 200, 100, 100, 200, 120\}$ and $\{0.001, 0.01, 0.01, 0.0001, 0.001, 0.001, 0.0001\}$, respectively; the list size of LBPR is set to 2; the window size of HCT is set to 2 and the weights of categories at layer 1 and layer 2 are 0.2 and 0.8 which are selected in [0,1] stepped by 0.1. Furthermore, to illustrate the robustness of the results, we have run the experiments under the same setup with different random seeds ten times and reported the overall performances by mean±standard_deviation in Tables 5.3 - 5.5.

5.2.2 Performance Comparison for CRS Methods (RQ1)

Table 5.3 presents the results of all CRS methods on SR@10 and AT across the four datasets. Fig. 5.2 presents the recommendation success rate (SR*@n) at different turns ($n = 1$ to 10), where SR* denotes the difference of each method with the strongest baseline SCPR and the SR*@n for SCPR is set to be zero for ease of presentation. As a whole, we can observe that our proposed CART achieves the best performance in comparison with state-of-the-art baselines. This firmly demonstrates the superiority of our proposed CART, that is, generating high-quality recommendation with less conversation turns.

Specifically, all methods start with SR=0, as they tend to accumulate a user's immediate preference (i.e., attributes) at the beginning of a conversation session. Max Entropy and Qrec generally underperform (RL) based baselines (CRM, EAR, SCPR and CART), which indicates that learning a better conversational strategy benefits the action selection, i.e., whether to ask an attribute or make recommendations. EAR and SCPR perform better than CRM, implying the efficacy of encoding both user preference and conversation history for the state vector of the policy network. Our CART outperforms the strongest baseline SCPR, mainly due to three reasons: **(1)** it delicately models both historical sequential behaviors and immediate preference via a translation-based method, which benefits the next POI recommendation by absorbing rich context information and overcoming the limitation of the FM-based recommender; **(2)** it designs an auxiliary reward when recommending a desired collective POI to well accommodate the fact that users usually prefer collective POIs with more choices; and **(3)** it adopts the rating-based sampling strategy for recommending next individual POIs within the desired collective POI in a fair way (i.e., making such POIs with low ratings have a chance to be recommended).

5.2.3 Performance Comparison for Static Methods (RQ2)

We compare the static methods with our proposed CART on the four datasets evaluated by Recall@K and MRR@K (K = 10), We follow the setting in [35] to select individual POIs within a predicted collective POI via the ratings for these static methods. Regarding CART, we examine its performance under different conversation turns from 1 to 10. The results are shown in Fig. 5.3, where we use two y-axises to visualize the results of the static methods (i.e., bar) and the proposed CART (i.e., blue line), respectively. Note that the scale of left y-axis is generally in the range of $(0, 0.08)$, whilst the right one is in the rage of $(0, 0.9)$.

Our CART significantly improves the performance comparing with these static methods, which implies the effectiveness of dynamic interaction (i.e., conversation module) for more accurate recommendations. Specifically, it far exceeds the runner up iMTL by 10

Table 5.3: Performance (represented by means and standard deviations) of all methods on the four datasets measured by SR@10 and AT, where the best performance is boldfaced; and the runner up is underlined.

	CAL		CHA		PHO		SIN	
	SR@10	AT	SR@10	AT	SR@10	AT	SR@10	AT
Max Entropy	0.686±0.003	6.13±0.636	0.501±0.001	6.31±0.671	0.503±0.007	6.30±0.836	0.427±0.003	6.89±0.772
CRM	0.699±0.024	4.57±0.511	0.466±0.017	4.59±0.782	0.495±0.052	4.31±0.935	0.634±0.014	5.61±0.823
Qrec	0.702±0.011	5.40±0.220	0.675±0.003	4.56±0.669	0.742±0.013	6.04±0.610	0.765±0.005	5.70±0.440
EAR	0.715±0.019	4.53±1.040	0.691±0.008	4.56±0.653	0.721±0.022	4.29±0.682	0.806±0.009	4.46±0.545
SCPR	<u>0.884</u>±0.010	<u>3.42</u>±0.340	<u>0.922</u>±0.012	<u>2.97</u>±0.722	<u>0.963</u>±0.002	<u>3.65</u>±0.656	<u>0.907</u>±0.012	<u>4.29</u>±0.633
CART	**0.937±0.010**	**3.13±0.500**	**0.948±0.012**	**2.54±0.978**	**0.975±0.004**	**3.58±0.891**	**0.922±0.012**	**4.14±0.642**

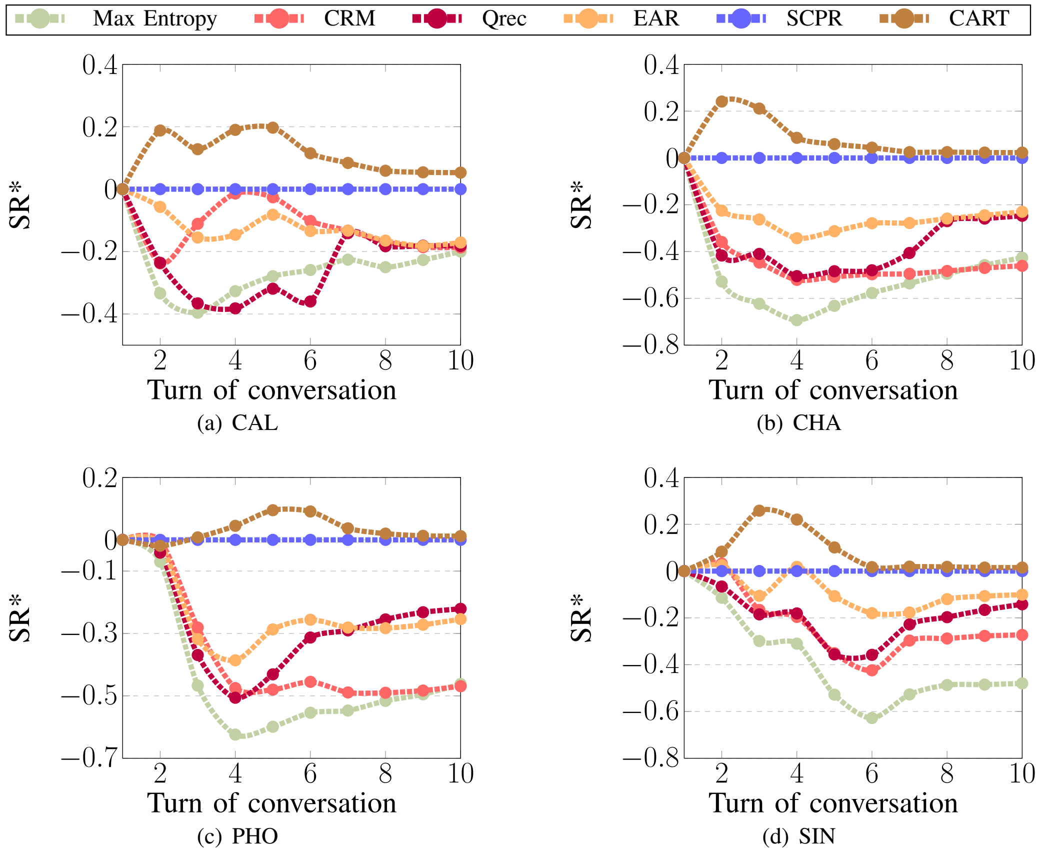
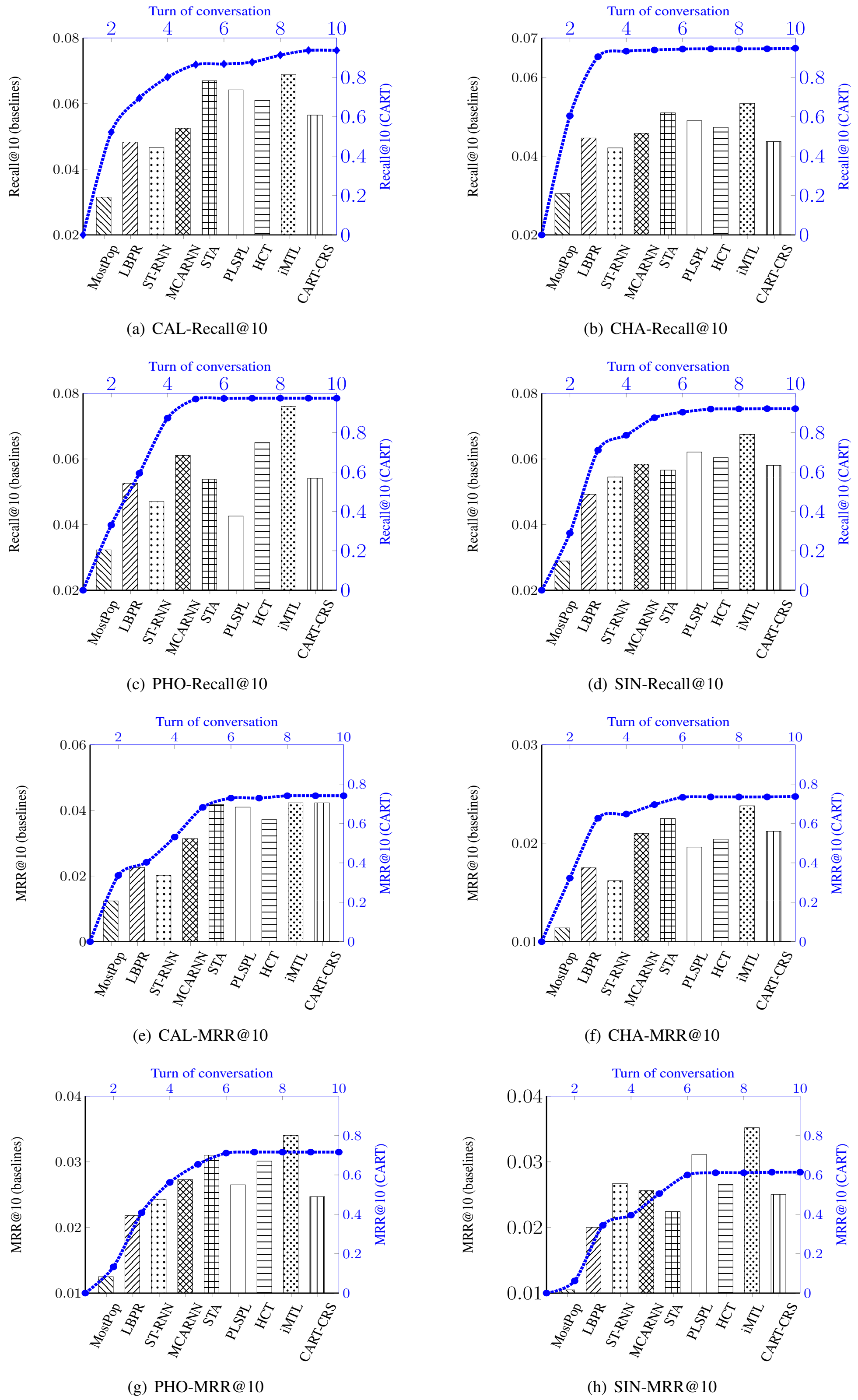


Fig. 5.2: Success Rate* (SR*@n) of CRS methods at each conversation turn. SR*@n refers to the difference of each method w.r.t SCPR.

and 17 times w.r.t. MRR@10 and Recall@10 on average within 3 turns. That is, CART is able to achieve better next POI recommendations after clarifying the user immediate preference (i.e., attributes) within 3 turns. It shows the great merit brought by CRS to the location-based recommendation community. For the comparison of static methods, baselines (MostPop, ST-RNN and PLSPL) ignoring the user activity (i.e., category of POIs) generally perform worse than those considering the user activity (e.g., LBPR, MCARNN, HCT, iMTL and CART-CRS). This verifies the efficacy of modeling a user's activity in next POI recommendation. Both STA and CART-CRS generally outperform LBPR, which suggests the superior generalization ability of the translation-based method for better recommendation. Moreover, regarding the multi-task based baselines, iMTL performs better than MCARNN, as it is tailored to model a user's uncertain activity and perform POI type prediction, thus helping more accurate activity and POI prediction.



(a) CAL-Recall@10
(b) CHA-Recall@10
(c) PHO-Recall@10
(d) SIN-Recall@10
(e) CAL-MRR@10
(f) CHA-MRR@10
(g) PHO-MRR@10
(h) SIN-MRR@10

Fig. 5.3: Performance comparison between static recommendation baselines and the proposed CART.

Table 5.4: Performance comparison of the CART and CART-*rand* with fixed maximum turn and random turn, respectively.

	CART		CART-*rand*	
	SR@10	AT	SR@$\hat{N}$	AT
CAL	0.937±0.010	3.13±0.500	0.725±0.005	3.46±0.420
CHA	0.948±0.012	2.54±0.978	0.763±0.003	2.76±0.383
PHO	0.975±0.004	3.58±0.891	0.804±0.003	2.52±0.579
SIN	0.922±0.012	4.14±0.642	0.752±0.016	4.23±0.614

5.2.4 Study on Quality of User Response (RQ3)

In this section, we aim to evaluate the impact of the quality of user response on our proposed CART via simulating a user's random interaction behaviors in the testing process[8]. The user simulator is widely employed to simulate conversation sessions [25,30,31]. However, such user simulator is built upon an assumption: the user would clearly express his preferences by responding to each question (i.e., asked attributes and recommendations) at each conversation turn until he accepts the recommendations or chooses to quit (i.e., reaching the maximum turn). There is no denying that such simulation has many limitations, but it is the most practical way at the current offline research stage [31].

However, in the real application scenario, one major issue is that the user may randomly interact with the system during the conversation sessions, instead of keeping interactions as stated in the above assumption. We, therefore, consider the following three cases in the testing process for a more comprehensive study: (1) *no interactions*: a user does not respond to either the asked attributes or recommendations; (2) *random interactions*: a user randomly interacts with the system, for example, he may respond only once to the asked attributes or recommendations, thus it is hard to identify a user's preference or know whether the recommendations meet his need; (3) *full interactions*: a user keeps interacting with the system until he accepts the recommendation or chooses to quit.

In this study, our focus is to build a location-aware CRS by employing a user simulator following the state-of-the-art [30] under case (3), for seeking the best strategy of action selection, so as to achieve more accurate next POI recommendations in fewer conversation turns. Hence, our work serves as an upper bound study of practical applications as we do

[8]We do not simulate a user's random interaction behaviors in our model training, because we can control the data quality in the training process.

Table 5.5: Performance (represented by means and standard deviations) of different variants of CART w.r.t. SR and AT across the four datasets, where the best performance is boldfaced; and the runner up is underlined.

	CAL				CHA			
	SR@5	SR@7	SR@10	AT	SR@5	SR@7	SR@10	AT
CART-$\mathbf{r}^{seq}$	0.228±0.045	0.515±0.042	0.918±0.005	7.45±0.442	0.367±0.023	0.527±0.016	0.909±0.019	6.09±0.863
CART-r_attn	0.228±0.040	0.339±0.034	0.632±0.004	8.55±0.450	0.887±0.006	0.891±0.003	0.893±0.024	3.15±0.901
CART(FM)	0.415±0.063	0.479±0.020	0.778±0.009	6.24±0.604	0.356±0.036	0.472±0.013	0.847±0.021	6.15±0.828
CART-$\mathbf{v}_{pre}$	**0.894±0.038**	**0.912±0.012**	$\underline{0.929}$±0.005	4.39±0.497	0.684±0.014	0.921±0.018	0.946±0.020	4.31±0.934
CART-$\mathbf{v}_{his}$	0.707±0.094	0.784±0.007	0.865±0.005	5.55±0.546	0.790±0.023	0.915±0.004	0.946±0.009	3.97±0.905
CART-$\mathbf{v}_{cand}$	0.818±0.063	0.847±0.003	0.865±0.010	3.77±0.281	$\underline{0.925}$±0.010	$\underline{0.943}$±0.011	**0.951±0.018**	2.75±0.832
CART-neg	0.679±0.068	0.790±0.010	0.825±0.007	4.62±0.345	0.239±0.007	0.488±0.005	0.745±0.013	6.57±0.770
CART-$auxi$	0.846±0.026	0.870±0.003	0.914±0.005	3.42±0.458	0.893±0.010	0.920±0.012	0.943±0.010	2.64±0.855
CART-$samp$	0.789±0.020	0.824±0.004	0.836±0.011	$\underline{3.27}$±0.530	0.764±0.015	0.767±0.020	0.769±0.022	$\underline{2.56}$±0.966
CART	$\underline{0.865}$±0.036	$\underline{0.877}$±0.006	**0.937±0.010**	**3.13±0.500**	**0.939±0.011**	**0.946±0.014**	$\underline{0.948}$±0.012	**2.54±0.978**

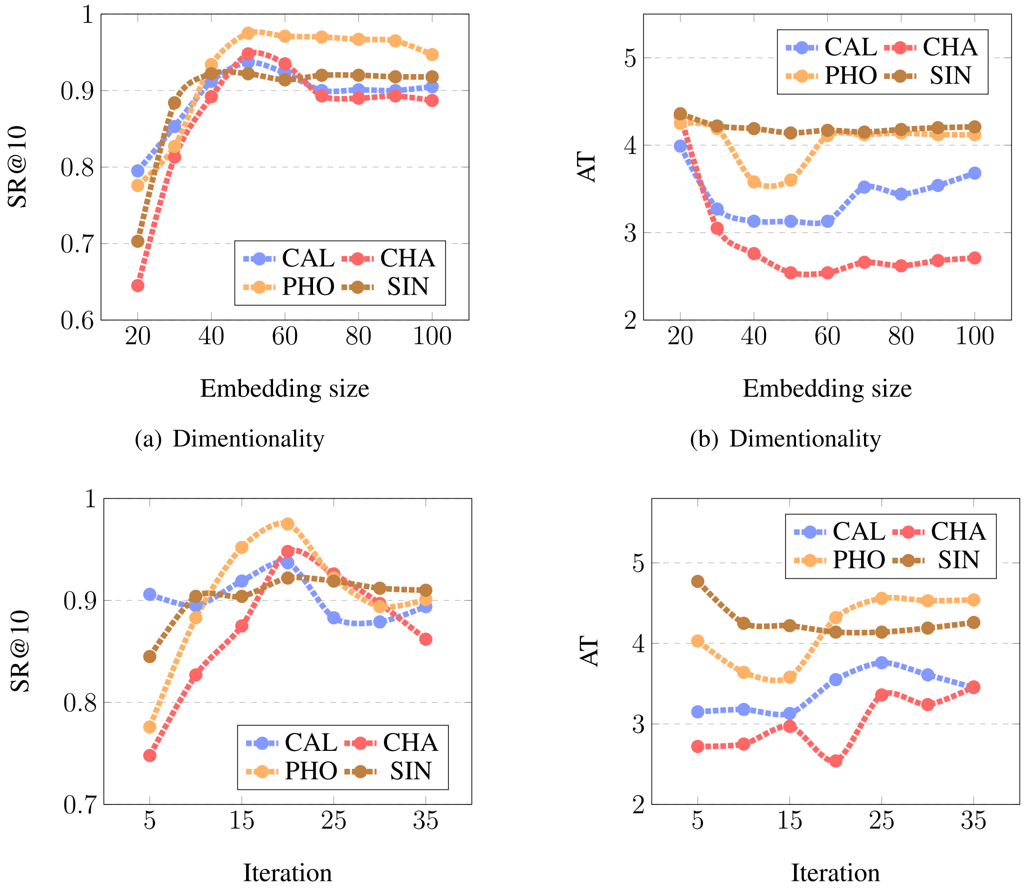
	PHO				SIN			
	SR@5	SR@7	SR@10	AT	SR@5	SR@7	SR@10	AT
CART-$\mathbf{r}^{seq}$	0.845±0.014	0.948±0.010	0.970±0.013	4.41±0.790	0.821±0.030	0.868±0.026	0.903±0.010	5.26±0.800
CART-r_attn	0.651±0.010	0.802±0.017	0.949±0.014	4.67±0.811	0.816±0.022	0.872±0.011	0.895±0.008	4.61±0.762
CART(FM)	0.705±0.004	0.725±0.004	0.753±0.002	4.34±0.616	0.637±0.016	0.753±0.005	0.858±0.011	5.39±0.804
CART-$\mathbf{v}_{pre}$	0.547±0.010	0.831±0.015	0.971±0.011	5.57±0.752	$\underline{0.838}$±0.035	0.880±0.006	$\underline{0.914}$±0.012	$\underline{4.34}$±0.647
CART-$\mathbf{v}_{his}$	$\underline{0.908}$±0.008	$\underline{0.972}$±0.010	**0.978±0.005**	$\underline{3.61}$±0.704	0.774±0.012	0.863±0.010	0.887±0.008	4.62±0.755
CART-$\mathbf{v}_{cand}$	0.653±0.024	0.820±0.020	0.946±0.012	4.80±0.733	0.767±0.027	0.881±0.018	0.913±0.014	4.93±0.813
CART-neg	0.489±0.012	0.625±0.004	0.764±0.005	4.58±0.845	0.771±0.013	0.825±0.010	0.848±0.006	5.01±0.688
CART-$auxi$	0.866±0.012	0.955±0.008	0.962±0.005	3.90±0.725	0.822±0.010	0.885±0.012	0.910±0.012	4.57±0.720
CART-$samp$	0.721±0.018	0.741±0.012	0.748±0.010	3.82±0.633	0.786±0.010	$\underline{0.890}$±0.015	0.892±0.014	4.88±0.705
CART	**0.971±0.010**	**0.973±0.004**	$\underline{0.975}$±0.004	**3.58±0.891**	**0.876±0.020**	**0.920±0.010**	**0.922±0.012**	**4.14±0.642**

not consider the real users' random behaviors during the interaction process. However, our proposed CART also works in cases (1) and (2). Specifically, in case (1), CART will degenerate to the static recommender to provide recommendations merely based on a user's historical check-in behaviors (see 'Recommender Module' in Fig. 5.1). In case (2), to simulate a user's random interaction behaviors, we randomly sample a conversation turn $\hat{N}$ in the range of [1, 10]. That is, a user would randomly interact with the system by $\hat{N}$ turns instead of the fixed maximum turn N (i.e., $N = 10$).

Table 5.4 presents the comparison results of our CART in cases (2) and (3). Note that the performance on case (1), that is, the degenerated CRS (i.e., CART-CRS), is shown in the comparison of static recommendation baselines (see Fig. 5.3). From the table, we notice that overall CART performs better than CART-$rand$, as it is challenging for accurate recommendations within a very limited conversation turns (e.g., CART could recommend a correct POI with four turns, but it may fail when sampling a small conversation turn $\hat{N} = 2$). In addition, we observe that CART-$rand$ generally outperforms both CRM and Qrec that adopt the fixed maximum turn as indicated in Table 5.3. This suggests the effectiveness of CART on learning conversational strategy in action selection and thus achieving successful recommendations with fewer conversation turns.

5.2.5 Ablation Study (RQ4)

To examine the efficacy of different components of CART, we compare its different variants from four aspects: (1) ablating important components from the recommender module, where CART-$\mathbf{r}^{seq}$ removes the sequential relation; CART-r_atten omits the relation-level attention, but directly performs sum aggregation for r^{seq} and r^{imm}; and CART(FM) indicates that the translation-based method is replaced by the FM-based method as adopted in state-of-the-art EAR [30]; (2) ablating each component in the state vector, where CART-$\mathbf{v}_{pre}$ removes the vector of user preference; CART-$\mathbf{v}_{his}$ omits the vector of conversation history; and CART-$\mathbf{v}_{cand}$ discards the vector encoding the size of candidate POIs; (3) testing the impact of online update, where CART-neg implies that we do not update the recommender by the immediate rejected POIs in conversations as negative samples; (4) validating the efficacy of the designed auxiliary reward and rating-based



(a) Dimentionality (b) Dimentionality

(c) Convergence (d) Convergence

Fig. 5.4: Results of parameter sensitivity.

sampling strategy, where CART-*auxi* indicates that we do not consider the auxiliary reward in Eq.(5.13), and CART-*samp* means that we simply select individual POIs within a collective POI via higher rating as in [35].

The results are reported in Table 5.5, where the CART generally outperforms its variants. Following the above four aspects, several interesting findings can be noted. **(1)** The CART without considering $\mathbf{r}^{seq}$ or r_attn performs worse than itself, e.g., less SR at the beginning turns (SR@5 in CAL) and larger AT in CAL and CHA. This indicates modeling the user's sequential behaviors and considering the attentive impacts of different relations indeed boost the performance. Besides, CART achieves better performance than CART(FM) under the same CRS settings, which suggests the superior generalization ability of translation-based methods over FM-based methods. **(2)** By removing

$\mathbf{v}_{pre}$ or $\mathbf{v}_{his}$, the value of SR drops heavily at the several beginning turns on the four datasets, while gradually increases at the future turns, which naturally results in high AT (i.e., lengthy conversations). The possible reason is that without $\mathbf{v}_{pre}$ and $\mathbf{v}_{his}$ as prior knowledge, the system needs to ask more attributes before making a recommendation; meanwhile, the candidate POI length $\mathbf{v}_{cand}$ is important since it assists in deciding when to recommend. **(3)** The performance decrease of CART-*neg* on SR (with a drop of 30.2% on average) and AT (with a drop of 37.1% on average) indicating the advantage of online update; **(4)** CART-*auxi* and CART-*samp* underperform CART on SR but achieve comparable AT, which shows that the design of auxiliary reward and rating-based sampling strategy are beneficial for a better next POI recommendation.

5.2.6 Hyper-parameter Analysis (RQ5)

Fig. 5.4 reports results regarding the parameter sensitivity on SR@10 and AT across the four datasets. Figs. 5.4(a-b) present the performance of CART with varying embedding size (other optimal hyper-parameters fixed), where the best performance achieves with the embedding size around 50. Figs. 5.4(c-d) show the convergence property of the CART, and we observe that it can converge within 20 iterations on the four datasets.

5.3 Summary

The conversational recommendation system has shown great potential in resolving the limitation of static recommendation methods. In this chapter, we propose a novel CART, which is the first work to commence conversation-based location service over uncertain check-ins. In particular, the adaptive relational translation based recommender aims to model a user's historical sequential behaviors and immediate preference received from conversations, and the conversation module seeks the conversational strategy for action selection and achieves successful recommendations with fewer conversation turns. Meanwhile, we design the auxiliary reward and rating-based sampling strategy to delicately consider the presence of collective POI in our scenario. Experimental results on real-world datasets show the superiority of CART on providing more accurate next POI recommendation over uncertain check-ins.

Chapter 6

Conclusions and Future Work

In this chapter, we first summarize the contributions of this dissertation, and then discuss several promising directions for future work.

6.1 Conclusions

The large amount of check-in trajectories delivered by users through the location-based services has greatly promoted the research in the next POI recommendation. Most existing next POI recommenders assume that these check-ins reflect users' real visits (i.e., certain check-ins), ignoring the presence of uncertain check-ins, which hinders their ability in dealing with the uncertain activities and incomplete context information. Therefore, in this dissertation, we aim to recommend next POIs for users with uncertain check-ins by exploring context information from their historical check-in behaviors. In particular, we attempt to investigate three recommendation problems, i.e., hierarchical category transition based recommendation, multi-task learning based recommendation and conversation-based recommendation.

Category transitions and category hierarchies are help to capture the insights of user's transition pattern over locations and effective in resolving the cold start issue, respectively. Existing studies either exploit the category transitions or category hierarchies for enhanced recommendation. They, however, fail to jointly model both effects in a unified manner, since there are different category transitions over different layers of category hierarchies (CHs).

In Chapter 3, we, therefore, focus on the problem of hierarchical category transition based recommendation by exploiting the categorical context: (1) the category transitions at different layers of the CHs, and (2) the hierarchical dependencies between POIs (categories) and categories in the CHs. The proposed HCT framework is able to predict users' preferred categories of uncertain check-ins at collective POIs based on the hierarchical category transition over certain check-ins. Moreover, the hierarchical dependencies in the CHs could help capture better semantic relatedness and learn high-quality representations of both POIs and categories. We conduct extensive experiments on the three datasets and demonstrate the superiority of our proposed HCT over the state-of-the-art next POI recommendation methods.

Despite HCT can alleviate the issue of uncertain check-ins in the next POI recommendation, it suffers from the challenges of characterizing the user's underlying activities over uncertain check-ins and modeling the interplay between sequential activities and locations. Recently, multi-task learning (MTL) based recommendation models have shown efficacy in improving recommendation performance by leveraging correlations among multiple related tasks. However, we argure that the potential of MTL based recommenders for next POI recommendation has not been fully exploited. Existing MTL methods neglect to explicitly model the interplay between related tasks (i.e., activity and location prediction tasks) with considering the spatiotemporal context and sequential regularity.

In Chapter 4, we thus concentrate on the problem of multi-task learning based recommendation. To achieve full exploitation of MTL for more accurate next POI recommendations, we devise an interactive multi-task learning framework (iMTL) to further resolving the issue of uncertain check-ins. In particular, iMTL equips with a two-channel encoder, i.e., temporal-aware activity encoder and spatial-aware location preference encoder, to capture the transitions of activities and locations, whereby a fuzzy characterization strategy is proposed to better represent activity over uncertain check-inns. The task-specific decoder then interactively aggregates the latent representations of the two-channel encoder to perform both activity and location predictions. Experimental results on the three datasets show the superiority on iMTL against the state-of-the-art algorithms w.r.t next activity and location recommendation tasks.

Existing next POI recommendation approaches (i.e., static methods), including HCT and iMTL, focus on learning user's accurate preference for his next check-in by exploiting various context information from historical behaviors. However, such static methods suffer from the inherent limitation of capturing user's immediate preference (i.e., exact intent) and incorporating user's feedback regarding the recommendation results. With the development of conversational techniques, the conversational recommendation system (CRS) has brings great potential in resolving the limitation of static recommenders, which motivates us to exploit a CRS in the next POI recommendation over the scenario of uncertain check-inns.

In Chaper 5, we investigate the problem of conversation based recommendation. To further ease the issue of uncertain check-ins in the next POI recommendation, we build a novel location-aware CRS, namely conversation-based adaptive relational translation framework (CART), which is composed of a recommender module and conversation module. Specifically, the adaptive relational translation based recommender aims to model a user's historical sequential behaviors and immediate preference received from conversations, and the conversation module seeks the conversational strategy for action selection and achieves successful recommendations with fewer conversations. Meanwhile, we design the auxiliary reward and rating-based sampling strategy to delicately consider the the presence of collective POIs in our recommendation scenario. Experiments on real-world datasets show the superiority of CART on providing more accurate next POI recommendation over uncertain check-ins.

To sum up, in this dissertation, we develop a series of recommendation approaches by exploiting context information from users' check-in behaviors, to resolve the problem of next POI recommendation with uncertain check-ins. Extensive experiments demonstrate the effectiveness of our proposed methods.

6.2 Future Work

In this dissertation, we mainly concentrate on exploiting various context features to resolve the issue of uncertain check-ins in the next POI recommendation. Based on the current studies, we present two promising directions to improve the recommendation performance for future work.

Data Augmentation for Improving Recommendation. Existing next POI recommendation methods, a.k.a sequential POI recommendation (SPR), are built upon the observed historical trajectories, e.g., $S = \{l_{t_1}, l_{t_2}, \ldots, l_{t_k}\}$. They aim to model user's historical sequential behaviors and predict a POI that user would like to visit at the next timestamp t_{k+1}, i.e., $P(l_{t_{k+1}}|S)$. In recent years, RNN and its variants have been extensively explored in SPR task, due to its promising ability in modeling sequential information. Despite the success of such RNN based models in next POI recommendation, they still suffer from the sparsity of user-location interactions. Intuitively, it is unlikely for a user to leave a check-in record every time when he visit a POI. As such, the sequential check-ins exploited in the current SPR models may lose some possible check-in records (e.g., l^*) between any successive location visits, e.g., $l_1 \rightarrow l^* \rightarrow l_2$, which impedes the ability of such methods in inferring accurate user preference. Therefore, we would like to complement the possible missing check-in behaviors (i.e., inferring possible activities and POIs) by taking advantages of sequential data augmentation techniques [109], to boost the next POI recommendation over the uncertain check-in scenario.

Inferring Future Preference for Improving Recommendation. With the assumption that users' next location visits are highly correlated with their latest check-in behaviors, some studies focus on modeling users' short-term behaviors, e.g., $S^{\mathbf{short}} = \{l_{t_1}, l_{t_2}, \ldots, l_{t_k}\}$, for next POI prediction, i.e., $P(l_{t_{k+1}}|S^{\text{short}})$. Other recent efforts consider to fuse both long-term and short-term sequential behaviors to achieve state-of-the-art performance. Generally, existing SPR methods mainly model user's past sequential behaviors to infer the preference of his next movement. In fact, a user's next movement (e.g., $l_{t_{k+1}}$) may be affected by his future POI visits (e.g., $l_{t_{k+2}}$) in addition to his past trajectories, i.e., S^{short} and S^{short}, since he may have activity planning in mind. In recent years, a few studies on sequential item recommendation tend to enhance recommendation performance by fusing explicit or implicit future signals [110–112]. To this end, in the static next POI recommendation scenario, although it is always impossible to acquire users' explicit future behaviors, we attempt to infer their implicit future potential preference derived from users' historical behaviors. In the conversational recommendation scenario, we are able to obtain user's explicit future behaviors/demands, such as shopping and going to a cinema, through the real time interactions. Hence, the CRS could not only improve the next POI recommendation performance but also recommend a trip plan regarding user's next sequential POI visits.

List of Publications

- **Lu Zhang**, Zhu Sun, Jie Zhang, Horst Kloeden and Felix Klanner. Modeling hierarchical category transition for next POI recommendation with uncertain check-ins. Information Sciences, 2020.

- **Lu Zhang**, Zhu Sun, Jie Zhang, Yu Lei, Chen Li, Ziqing Wu, Horst Kloeden and Felix Klanner. An Interactive Multi-Task Learning Framework for Next POI Recommendation with Uncertain Check-ins. Proceedings of the 29th International Joint Conference on Artificial Intelligence (IJCAI), 2020.

- Zhu Sun, Chen Li, Yu Lei, **Lu Zhang**, Jie Zhang, and Shunpan Liang. Point-of-Interest Recommendation for Users-Businesses with Uncertain Check-ins. IEEE Transactions on Knowledge and Data Engineering (TKDE), 2021.

- **Lu Zhang**, Zhu Sun, Jie Zhang, Yiwen Wu and Yunwen Xia. Conversation-based Adaptive Relational Translation Method for Next POI Recommendation with Uncertain Check-ins. IEEE Transactions on Neural Networks and Learning Systems (TNNLS), 2022.

- **Lu Zhang**, Zhu Sun, Ziqing Wu, Jie Zhang, Yew Soon Ong and Xinghua Qu. Next Point-of-Interest Recommendation with Inferring Multi-step Future Preference. Proceedings of the 31st International Joint Conference on Artificial Intelligence (IJCAI), 2022.

List of Submissions

- Zhu Sun, Yu Lei, **Lu Zhang**, Chen Li, Yew Soon Ong, Jie Zhang. A Multi-Channel Next POI Recommendation Framework with Multi-Granularity Check-in Signals. 31st ACM International Conference on Information and Knowledge Management (CIKM), 2022.

References

[1] B. Liu, Y. Fu, Z. Yao, and H. Xiong, “Learning geographical preferences for point-of-interest recommendation,” in *Proceedings of the 19th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 2013, pp. 1043–1051.

[2] 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.

[3] C. Cheng, H. Yang, M. R. Lyu, and I. King, “Where you like to go next: Successive point-of-interest recommendation,” in *Proceedings of the 23th International Joint Conference on Artificial Intelligence*, 2013, pp. 2605–2611.

[4] S. Feng, X. Li, Y. Zeng, G. Cong, Y. M. Chee, and Q. Yuan, “Personalized ranking metric embedding for next new poi recommendation,” in *Proceedings of the 24th International Joint Conference on Artificial Intelligence*, 2015, pp. 2069—-2075.

[5] S. Zhao, T. Zhao, H. Yang, M. R. Lyu, and I. King, “Stellar: spatial-temporal latent ranking for successive point-of-interest recommendation,” in *Proceedings of the 30th AAAI Conference on Artificial Intelligence*, 2016, pp. 315–321.

[6] D. Yao, C. Zhang, J. Huang, and J. Bi, “Serm: A recurrent model for next location prediction in semantic trajectories,” in *Proceedings of the 2017 ACM on Conference on Information and Knowledge Management*, 2017, pp. 2411–2414.

[7] D. Liao, W. Liu, Y. Zhong, J. Li, and G. Wang, “Predicting activity and location with multi-task context aware recurrent neural network.” in *Proceedings of the 27th International Joint Conference on Artificial Intelligence*, 2018, pp. 3435–3441.

[8] P. Zhao, H. Zhu, Y. Liu, J. Xu, Z. Li, F. Zhuang, V. S. Sheng, and X. Zhou, "Where to go next: a spatio-temporal gated network for next poi recommendation," in *Proceedings of the 33rd AAAI Conference on Artificial Intelligence*, 2019, pp. 5877–5884.

[9] J. He, X. Li, and L. Liao, "Category-aware next point-of-interest recommendation via listwise bayesian personalized ranking." in *Proceedings of the 26th International Joint Conference on Artificial Intelligence*, 2017, pp. 1837–1843.

[10] J. He, X. Li, L. Liao, D. Song, and W. K. Cheung, "Inferring a personalized next point-of-interest recommendation model with latent behavior patterns," in *Proceedings of the 30th AAAI Conference on Artificial Intelligence*, 2016.

[11] X. Liu, Y. Liu, K. Aberer, and C. Miao, "Personalized point-of-interest recommendation by mining users' preference transition," in *Proceedings of the 22nd ACM International Conference on Information and Knowledge Management*, 2013, pp. 733–738.

[12] Y. Li, R. Zheng, T. Tian, Z. Hu, R. Iyer, and K. Sycara, "Joint embedding of hierarchical categories and entities for concept categorization and dataless classification," in *Proceedings of the 26th International Conference on Computational Linguistics*, 2016, pp. 2678–2688.

[13] J. Yang, Z. Sun, A. Bozzon, and J. Zhang, "Learning hierarchical feature influence for recommendation by recursive regularization," in *Proceedings of the 10th ACM Conference on Recommender Systems*, 2016, pp. 51–58.

[14] Z. Sun, J. Yang, J. Zhang, and A. Bozzon, "Exploiting both vertical and horizontal dimensions of feature hierarchy for effective recommendation," in *Proceedings of the 31st AAAI Conference on Artificial Intelligence*, 2017, pp. 189–195.

[15] Z. Sun, Q. Guo, J. Yang, H. Fang, G. Guo, J. Zhang, and R. Burke, "Research commentary on recommendations with side information: A survey and research directions," *Electronic Commerce Research and Applications*, vol. 37, p. 100879, 2019.

[16] J. Zhang, L. Nie, X. Wang, X. He, X. Huang, and T. S. Chua, "Shorter-is-better: venue category estimation from micro-video," in *Proceedings of the 24th ACM International Conference on Multimedia*, 2016, pp. 1415–1424.

[17] M. Liu, L. Nie, X. Wang, Q. Tian, and B. Chen, "Online data organizer: micro-video categorization by structure-guided multimodal dictionary learning," *IEEE Transactions on Image Processing*, vol. 28, no. 3, pp. 1235–1247, 2018.

[18] R. Caruana, "Multitask learning," *Machine learning*, vol. 28, no. 1, pp. 41–75, 1997.

[19] I. Sutskever, O. Vinyals, and Q. V. Le, "Sequence to sequence learning with neural networks," in *Advances in neural information processing systems*, 2014, pp. 3104–3112.

[20] J. Rao, F. Ture, and J. Lin, "Multi-task learning with neural networks for voice query understanding on an entertainment platform," in *Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 2018, pp. 636–645.

[21] Y. Lu, R. Dong, and B. Smyth, "Why I like it: multi-task learning for recommendation and explanation," in *Proceedings of the 12th ACM Conference on Recommender Systems*, 2018, pp. 4–12.

[22] Z. Chen, X. Wang, X. Xie, T. Wu, G. Bu, Y. Wang, and E. Chen, "Co-attentive multi-task learning for explainable recommendation," in *Proceedings of the 28th International Joint Conference on Artificial Intelligence*, 2019, pp. 2137–2143.

[23] Z. Zhao, L. Hong, L. Wei, J. Chen, A. Nath, S. Andrews, A. Kumthekar, M. Sathiamoorthy, X. Yi, and E. Chi, "Recommending what video to watch next: a multi-task ranking system," in *Proceedings of the 13th ACM Conference on Recommender Systems*, 2019, pp. 43–51.

[24] J. Zhong, C. Ma, J. Zhou, and W. Wang, "From when to where: A multi-task learning approach for next point-of-interest recommendation," in *International Conference on Wireless Algorithms, Systems, and Applications*, 2020, pp. 781–793.

[25] Y. Sun and Y. Zhang, "Conversational recommender system," in *Proceedings of the 41st International ACM SIGIR Conference on Research and Development in Information Retrieval*, 2018, pp. 235–244.

[26] Y. Zhang, X. Chen, Q. Ai, L. Yang, and W. B. Croft, "Towards conversational search and recommendation: system ask, user respond," in *Proceedings of the 27th ACM International Conference on Information and Knowledge Management*, 2018, pp. 177–186.

[27] B. Priyogi, "Preference elicitation strategy for conversational recommender system," in *Proceedings of the 12th ACM International Conference on Web Search and Data Mining*, 2019, pp. 824–825.

[28] J. Zou, Y. Chen, and E. Kanoulas, "Towards question-based recommender systems," in *Proceedings of the 43rd International ACM SIGIR Conference on Research and Development in Information Retrieval*, 2020, pp. 881–890.

[29] S. Li, W. Lei, Q. Wu, X. He, P. Jiang, and T.-S. Chua, "Seamlessly unifying attributes and items: Conversational recommendation for cold-start users," *arXiv preprint arXiv:2005.12979*, 2020.

[30] W. Lei, X. He, Y. Miao, Q. Wu, R. Hong, M.-Y. Kan, and T.-S. Chua, "Estimation-action-reflection: Towards deep interaction between conversational and recommender systems," in *Proceedings of the 13th International Conference on Web Search and Data Mining*, 2020, pp. 304–312.

[31] W. Lei, G. Zhang, X. He, Y. Miao, X. Wang, L. Chen, and T.-S. Chua, "Interactive path reasoning on graph for conversational recommendation," in *Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 2020, pp. 2073–2083.

[32] L. Zhang, Z. Sun, J. Zhang, H. Kloeden, and F. Klanner, "Modeling hierarchical category transition for next poi recommendation with uncertain check-ins," *Information Sciences*, vol. 515, pp. 169–190, 2020.

[33] Q. Liu, S. Wu, L. Wang, and T. Tan, "Predicting the next location: a recurrent model with spatial and temporal contexts," in *Proceedings of the 30th AAAI Conference on Artificial Intelligence*, 2016, pp. 194–200.

[34] L. Huang, Y. Ma, S. Wang, and Y. Liu, "An attention-based spatiotemporal lstm network for next poi recommendation," *IEEE Transactions on Services Computing*, pp. 1–1, 2019.

[35] L. Zhang, Z. Sun, J. Zhang, Y. Lei, C. Li, Z. Wu, H. Kloeden, and F. Klanner, "An interactive multi-task learning framework for next poi recommendation with uncertain check-ins," in *Proceedings of the 29th International Joint Conference on Artificial Intelligence*, 2020, pp. 3551–3557.

[36] S. Rendle, "Factorization machines," in *2010 IEEE International Conference on Data Mining*, 2010, pp. 995–1000.

[37] T. Mikolov, K. Chen, G. Corrado, and J. Dean, "Efficient estimation of word representations in vector space," *arXiv preprint arXiv:1301.3781*, 2013.

[38] S. Hochreiter and J. Schmidhuber, "Long short-term memory," *Neural Computation*, vol. 9, no. 8, pp. 1735–1780, 1997.

[39] R. Girshick, "Fast R-CNN," in *Proceedings of the IEEE International Conference on Computer Vision*, 2015, pp. 1440–1448.

[40] T. Qian, B. Liu, Q. V. H. Nguyen, and H. Yin, "Spatiotemporal representation learning for translation-based poi recommendation," *ACM Transactions on Information Systems*, vol. 37, no. 2, pp. 1–24, 2019.

[41] F. Yu, L. Cui, W. Guo, X. Lu, Q. Li, and H. Lu, "A category-aware deep model for successive poi recommendation on sparse check-in data," in *Proceedings of the 29th International World Wide Web Conference*, 2020, pp. 1264–1274.

[42] H. Wang, H. Shen, W. Ouyang, and X. Cheng, "Exploiting poi-specific geographical influence for point-of-interest recommendation." in *Proceedings of the 27th International Joint Conference on Artificial Intelligence*, 2018, pp. 3877–3883.

[43] H. A. Rahmani, M. Aliannejadi, M. Baratchi, and F. Crestani, "Joint geographical and temporal modeling based on matrix factorization for point-of-interest recommendation," in *European Conference on Information Retrieval*, 2020, pp. 205–219.

[44] Y. Lin, H. Wan, S. Guo, and Y. Lin, "Pre-training context and time aware location embeddings from spatial-temporal trajectories for user next location prediction," in *Proceedings of the 34th AAAI Conference on Artificial Intelligence*, 2020.

[45] Z. Sun, C. Li, Y. Lei, L. Zhang, J. Zhang, and S. Liang, "Point-of-interest recommendation for users-businesses with uncertain check-ins," *IEEE Transactions on Knowledge and Data Engineering*, 2021.

[46] Y. Luo, Q. Liu, and Z. Liu, "Stan: Spatio-temporal attention network for next location recommendation," in *Proceedings of the Web Conference*, 2021, pp. 2177–2185.

[47] W. R. Tobler, "A computer movie simulating urban growth in the detroit region," *Economic geography*, vol. 46, no. sup1, pp. 234–240, 1970.

[48] M. Ye, P. Yin, W.-C. Lee, and D.-L. Lee, "Exploiting geographical influence for collaborative point-of-interest recommendation," in *Proceedings of the 34th ACM SIGIR International Conference on Research and development in Information Retrieval*, 2011, pp. 325–334.

[49] Q. Yuan, G. Cong, Z. Ma, A. Sun, and N. M. Thalmann, "Time-aware point-of-interest recommendation," in *Proceedings of the 36th International ACM SIGIR Conference on Research and Development in Information Retrieval*, 2013, pp. 363–372.

[50] X. Li, G. Cong, X.-L. Li, T.-A. N. Pham, and S. Krishnaswamy, "Rank-geofm: a ranking based geographical factorization method for point of interest recommendation," in *Proceedings of the 38th International ACM SIGIR Conference on Research and Development in Information Retrieval*, 2015, pp. 433–442.

[51] C. Liu, J. Liu, J. Wang, S. Xu, H. Han, and Y. Chen, "An attention-based spatiotemporal gated recurrent unit network for point-of-interest recommendation," *ISPRS International Journal of Geo-Information*, vol. 8, no. 8, p. 355, 2019.

[52] Y. Liu, W. Wei, A. Sun, and C. Miao, "Exploiting geographical neighborhood characteristics for location recommendation," in *Proceedings of the 23rd ACM International Conference on Information and Knowledge Management*, 2014, pp. 739–748.

[53] S. Feng, G. Cong, B. An, and Y. M. Chee, "Poi2vec: Geographical latent representation for predicting future visitors," in *Proceedings of the 31st AAAI Conference on Artificial Intelligence*, 2017, pp. 102–108.

[54] H. Luo, J. Zhou, Z. Bao, S. Li, J. S. Culpepper, H. Ying, H. Liu, and H. Xiong, "Spatial object recommendation with hints: When spatial granularity matters," in *Proceedings of the 43rd International ACM SIGIR Conference on Research and Development in Information Retrieval*, 2020, pp. 781–790.

[55] H. Gao, J. Tang, X. Hu, and H. Liu, "Exploring temporal effects for location recommendation on location-based social networks," in *Proceedings of the 7th ACM Conference on Recommender Systems*, 2013, pp. 93–100.

[56] Y. Wu, K. Li, G. Zhao, and Q. Xueming, "Personalized long-and short-term preference learning for next poi recommendation," *IEEE Transactions on Knowledge and Data Engineering*, pp. 1–1, 2020.

[57] B. Chang, Y. Park, D. Park, S. Kim, and J. Kang, "Content-aware hierarchical point-of-interest embedding model for successive poi recommendation." in *Proceedings of the 27th International Joint Conference on Artificial Intelligence*, 2018, pp. 3301–3307.

[58] J. Feng, Y. Li, C. Zhang, F. Sun, F. Meng, A. Guo, and D. Jin, "Deepmove: predicting human mobility with attentional recurrent networks," in *Proceedings of the 27th World Wide Web Conference*, 2018, pp. 1459–1468.

[59] Q. Guo, Z. Sun, J. Zhang, and Y.-L. Theng, “An attentional recurrent neural network for personalized next location recommendation,” in *Proceedings of the 34th AAAI Conference on Artificial Intelligence*, vol. 34, no. 01, 2020, pp. 83–90.

[60] K. Sun, T. Qian, T. Chen, Y. Liang, Q. V. H. Nguyen, and H. Yin, “Where to go next: Modeling long-and short-term user preferences for point-of-interest recommendation,” in *Proceedings of the 34th AAAI Conference on Artificial Intelligence*, vol. 34, no. 01, 2020, pp. 214–221.

[61] Y. Shi, M. Larson, and A. Hanjalic, “Collaborative filtering beyond the user-item matrix: a survey of the state of the art and future challenges,” *ACM Computing Surveys (CSUR)*, vol. 47, no. 1, p. 3, 2014.

[62] S. Rendle, C. Freudenthaler, and L. Schmidt-Thieme, “Factorizing personalized markov chains for next-basket recommendation,” in *Proceedings of the 19th International Conference on World Wide Web*, 2010, pp. 811–820.

[63] J.-D. Zhang, C.-Y. Chow, and Y. Li, “Lore: exploiting sequential influence for location recommendations,” in *Proceedings of the 22nd ACM SIGSPATIAL International Conference on Advances in Geographic Information Systems*, 2014, pp. 103–112.

[64] Z. Sun, J. Yang, J. Zhang, A. Bozzon, Y. Chen, and C. Xu, “MRLR: multi-level representation learning for personalized ranking in recommendation,” in *Proceedings of the 26th International Joint Conference on Artificial Intelligence*, 2017, pp. 2807–2813.

[65] O. Barkan and N. Koenigstein, “Item2vec: neural item embedding for collaborative filtering,” in *Proceedings of the 26th IEEE International Workshop on Machine Learning for Signal Processing*, 2016, pp. 1–6.

[66] T. Mikolov, I. Sutskever, K. Chen, G. S. Corrado, and J. Dean, “Distributed representations of words and phrases and their compositionality,” in *Proceedings of the Advances in Neural Information Processing Systems*, 2013, pp. 3111–3119.

[67] M. G. Ozsoy, "From word embeddings to item recommendation," *arXiv preprint arXiv:1601.01356*, 2016.

[68] X. Liu, Y. Liu, and X. Li, "Exploring the context of locations for personalized location recommendations," in *Proceedings of the 25th International Joint Conference on Artificial Intelligence*, 2016, pp. 1188–1194.

[69] 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*, 2017, pp. 153–162.

[70] J. Ye, Z. Zhu, and H. Cheng, "What's your next move: User activity prediction in location-based social networks," in *Proceedings of the 2013 SIAM International Conference on Data Mining*, 2013, pp. 171–179.

[71] D. Liao, Y. Zhong, and J. Li, "Location prediction through activity purpose: integrating temporal and sequential models," in *Pacific-Asia Conference on Knowledge Discovery and Data Mining*, 2017, pp. 711–723.

[72] N. Wang, H. Wang, Y. Jia, and Y. Yin, "Explainable recommendation via multitask learning in opinionated text data," in *The 41st International ACM SIGIR Conference on Research and Development in Information Retrieval*, 2018, pp. 165–174.

[73] J. Feng, Y. Li, Z. Yang, Q. Qiu, and D. Jin, "Predicting human mobility with semantic motivation via multi-task attentional recurrent networks," *IEEE Transactions on Knowledge and Data Engineering*, 2020.

[74] W. Liang and W. Zhang, "Learning social relations and spatiotemporal trajectories for next check-in inference," *IEEE Transactions on Neural Networks and Learning Systems*, pp. 1–11, 2020.

[75] K. Zhao, Y. Zhang, H. Yin, J. Wang, K. Zheng, X. Zhou, and C. Xing, "Discovering subsequence patterns for next poi recommendation." in *Proceedings of the 29th International Joint Conference on Artificial Intelligence*, 2020, pp. 3216–3222.

[76] C. Angulo, Z. Falomir, D. Anguita, N. Agell, and E. Cambria, "Bridging cognitive models and recommender systems," *Cognitive Computation*, vol. 12, no. 2, pp. 426–427, 2020.

[77] E. Cambria, D. Das, S. Bandyopadhyay, and A. Feraco, "Affective computing and sentiment analysis," in *A practical guide to sentiment analysis.* Springer, 2017, pp. 1–10.

[78] C. Gao, W. Lei, X. He, M. de Rijke, and T.-S. Chua, "Advances and challenges in conversational recommender systems: A survey," *arXiv preprint arXiv:2101.09459*, 2021.

[79] K. Christakopoulou, F. Radlinski, and K. Hofmann, "Towards conversational recommender systems," in *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 2016, pp. 815–824.

[80] C. Greco, A. Suglia, P. Basile, and G. Semeraro, "Converse-et-impera: Exploiting deep learning and hierarchical reinforcement learning for conversational recommender systems," in *Conference of the Italian Association for Artificial Intelligence.* Springer, 2017, pp. 372–386.

[81] K. Christakopoulou, A. Beutel, R. Li, S. Jain, and E. H. Chi, "Q&r: A two-stage approach toward interactive recommendation," in *Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 2018, pp. 139–148.

[82] L. Liao, Y. Ma, X. He, R. Hong, and T.-s. Chua, "Knowledge-aware multimodal dialogue systems," in *Proceedings of the 26th ACM International Conference on Multimedia*, 2018, pp. 801–809.

[83] R. Li, S. E. Kahou, H. Schulz, V. Michalski, L. Charlin, and C. Pal, "Towards deep conversational recommendations," in *32nd Conference on Neural Information Processing Systems*, 2018, pp. 9725–9735.

[84] T. Yu, Y. Shen, and H. Jin, "A visual dialog augmented interactive recommender system," in *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 2019, pp. 157–165.

[85] N. Sardella, C. Biancalana, A. Micarelli, and G. Sansonetti, "An approach to conversational recommendation of restaurants," in *International Conference on Human-Computer Interaction*, 2019, pp. 123–130.

[86] X. Zhang, H. Xie, H. Li, and J. Lui, "Toward building conversational recommender systems: A contextual bandit approach," *arXiv preprint arXiv:1906.01219*, 2019.

[87] Q. Chen, J. Lin, Y. Zhang, M. Ding, Y. Cen, H. Yang, and J. Tang, "Towards knowledge-based recommender dialog system," *arXiv preprint arXiv:1908.05391*, 2019.

[88] Z. Liu, H. Wang, Z.-Y. Niu, H. Wu, W. Che, and T. Liu, "Towards conversational recommendation over multi-type dialogs," *arXiv preprint arXiv:2005.03954*, 2020.

[89] K. Xu, J. Yang, J. Xu, S. Gao, J. Guo, and J.-R. Wen, "Adapting user preference to online feedback in multi-round conversational recommendation," in *Proceedings of the 14th ACM International Conference on Web Search and Data Mining*, 2021, pp. 364–372.

[90] X. Zhang, H. Xie, H. Li, and J. CS Lui, "Conversational contextual bandit: Algorithm and application," in *Proceedings of the 29th International World Wide Web Conference*, 2020, pp. 662–672.

[91] O. Chapelle and L. Li, "An empirical evaluation of thompson sampling," *Advances in Neural Information Processing Systems*, vol. 24, pp. 2249–2257, 2011.

[92] N. Mrkšić, D. O. Séaghdha, T.-H. Wen, B. Thomson, and S. Young, "Neural belief tracker: Data-driven dialogue state tracking," *arXiv preprint arXiv:1606.03777*, 2016.

[93] T. Young, E. Cambria, I. Chaturvedi, H. Zhou, S. Biswas, and M. Huang, "Augmenting end-to-end dialogue systems with commonsense knowledge," in *Proceedings of the 32nd AAAI Conference on Artificial Intelligence*, 2018, pp. 4970–4977.

[94] H. Xu, H. Peng, H. Xie, E. Cambria, L. Zhou, and W. Zheng, "End-to-end latent-variable task-oriented dialogue system with exact log-likelihood optimization," *World Wide Web*, vol. 23, no. 3, pp. 1989–2002, 2020.

[95] D. Yang, D. Zhang, and B. Qu, "Participatory cultural mapping based on collective behavior data in location-based social networks," *ACM Transactions on Intelligent Systems and Technology*, vol. 7, no. 3, p. 30, 2016.

[96] M. Xie, H. Yin, H. Wang, F. Xu, W. Chen, and S. Wang, "Learning graph-based poi embedding for location-based recommendation," in *Proceedings of the 25th ACM International on Conference on Information and Knowledge Management*, 2016, pp. 15–24.

[97] Y. Liu, T.-A. N. Pham, G. Cong, and Q. Yuan, "An experimental evaluation of point-of-interest recommendation in location-based social networks," *Proceedings of the VLDB Endowment*, vol. 10, no. 10, pp. 1010–1021, 2017.

[98] S. Rendle, C. Freudenthaler, Z. Gantner, and L. Schmidt-Thieme, "BPR: Bayesian personalized ranking from implicit feedback," in *Proceedings of the 25th Conference on Uncertainty in Artificial Intelligence*, 2009, pp. 452–461.

[99] J. Duchi, E. Hazan, and Y. Singer, "Adaptive subgradient methods for online learning and stochastic optimization." *Journal of Machine Learning Research*, vol. 12, no. 7, 2011.

[100] L. Zhang, Z. Sun, J. Zhang, Y. Wu, and Y. Xia, "Conversation-based adaptive relational translation method for next poi recommendation with uncertain check-ins," *IEEE Transactions on Neural Networks and Learning Systems*, 2022.

[101] A. Bordes, N. Usunier, A. Garcia-Duran, J. Weston, and O. Yakhnenko, "Translating embeddings for modeling multi-relational data," *Advances in Neural Information Processing Systems*, vol. 26, pp. 2787–2795, 2013.

[102] B. Wu, X. He, Z. Sun, L. Chen, and Y. Ye, "Atm: An attentive translation model for next-item recommendation," *IEEE Transactions on Industrial Informatics*, vol. 16, no. 3, pp. 1448–1459, 2019.

[103] X. Yang, Y. Ma, L. Liao, M. Wang, and T.-S. Chua, "Transnfcm: Translation-based neural fashion compatibility modeling," in *Proceedings of the 33rd AAAI Conference on Artificial Intelligence*, vol. 33, no. 01, 2019, pp. 403–410.

[104] Z. Wang, J. Zhang, J. Feng, and Z. Chen, "Knowledge graph embedding by translating on hyperplanes," in *Proceedings of the 28th AAAI Conference on Artificial Intelligence*, vol. 28, no. 1, 2014, pp. 1112–1119.

[105] Y. Lin, Z. Liu, M. Sun, Y. Liu, and X. Zhu, "Learning entity and relation embeddings for knowledge graph completion," in *Proceedings of the 29th AAAI Conference on Artificial Intelligence*, 2015, pp. 2181–2187.

[106] Z. Sun, Z.-H. Deng, J.-Y. Nie, and J. Tang, "Rotate: Knowledge graph embedding by relational rotation in complex space," *arXiv preprint arXiv:1902.10197*, 2019.

[107] R. S. Sutton, D. A. McAllester, S. P. Singh, and Y. Mansour, "Policy gradient methods for reinforcement learning with function approximation," in *Advances in Neural Information Processing Systems*, 2000, pp. 1057–1063.

[108] M. F. Dacrema, P. Cremonesi, and D. Jannach, "Are we really making much progress? a worrying analysis of recent neural recommendation approaches," in *Proceedings of the 13th ACM Conference on Recommender Systems*, 2019, pp. 101–109.

[109] Y. Li, Y. Luo, Z. Zhang, S. Sadiq, and P. Cui, "Context-aware attention-based data augmentation for poi recommendation," in *2019 IEEE 35th International Conference on Data Engineering Workshops*, 2019, pp. 177–184.

[110] E. Shao, S. Guo, and Z. A. Pardos, "Degree planning with plan-bert: Multi-semester recommendation using future courses of interest," in *Proceedings of the 35th AAAI Conference on Artificial Intelligence*, vol. 35, no. 17, 2021, pp. 14 920–14 929.

[111] F. Yuan, X. He, H. Jiang, G. Guo, J. Xiong, Z. Xu, and Y. Xiong, "Future data helps training: Modeling future contexts for session-based recommendation," in *Proceedings of the Web Conference*, 2020, pp. 303–313.

[112] Y. Lu, S. Zhang, Y. Huang, L. Wang, X. Yu, Z. Zhao, and F. Wu, "Future-aware diverse trends framework for recommendation," in *Proceedings of the Web Conference*, 2021, pp. 2992–3001.