DISSERTATION

Learning from Location Histories for Location Recommendations in LBS

Ausgeführt zum Zwecke der Erlangung des akademischen Grades eines Doktors der technischen Wissenschaften unter der Leitung von

Univ. Prof. Mag. rer. nat. Dr. rer. nat. Georg Gartner
E120-6
Department für Geodäsie und Geoinformation

Mitbetreuung:
Univ. Prof. Dr. Martin Raubal
Eidgenössische Technische Hochschule Zürich (ETH Zürich)

eingereicht an der Technischen Universität Wien

Fakultät für Mathematik und Geoinformation

von

Haosheng Huang
Matrikelnummer: 0728176
Scheugasse 4/6
1100 Wien

Wien, am 05.09.2013 ______________________

Abstract

When visiting a new city, tourists often need help to identify personally interesting places or locations from a potentially overwhelming set of choices. Recently, the increasing availability of GPS-enabled devices and the rapid advances in geotagged social media have led to the accumulation of a large amount of location histories, which may reflect people's travel experiences in the environment. Research has shown that experiences from past users (especially similar ones) in similar contexts can help the current users to solve their problems efficiently, e.g., choosing where to visit next.

Motivated by the above aspects, this thesis explores a methodology of deriving recommendations from location histories in Location Based Services (LBS). More specifically, we investigate how human location histories (e.g., GPS trajectories and trajectories constructed from Flickr photos) and collaborative filtering (CF) can be integrated in LBS to provide users with personalized and context-aware location recommendations. The main work and findings are summarized as follows.

1) In order to represent the information extracted from users' location histories, a model of contextual user profiles is developed based on the concepts of stops and moves, which have been shown to be a useful conceptual framework for processing raw location histories in the literature. The proposed model provides a uniform conceptual framework for representing users' interests in various locations (reflected by their visits to these locations and the duration of these visits) as well as their behavior of visiting such locations, as extracted from different types of location histories. Methods of extracting meaningful user profiles from raw location histories are also developed.
2) We investigate how information extracted from other users' location histories (i.e., their interests in various locations, and their behavior of visiting such locations) can be aggregated for providing the current user with personalized location recommendations. The evaluation shows that considering other people's movements, sequence relationships of locations visited, location popularity, duration at locations, and transit time between locations contributes to the improvement of recommendation quality. Among them, considering other

people's movements achieves the biggest improvement.

3) The personalized recommendation algorithm developed above is further improved by integrating additional contextual information, such as weather and companion (with whom). Specifically, we develop a methodology of identifying context parameters/dimensions that are relevant for making recommendations, and explore a context similarity measure. We then design three approaches to integrate the similarity measure into CF for making context-aware location recommendations. The evaluation demonstrates that: (a) When including contextual information into CF, choosing a suitable set of relevant context parameters is very important and may greatly affect the recommendation performance; the identification of a set of relevant context parameters can be achieved by analyzing how users' aggregated movements differ in different situations; (b) The contextual post-filtering method achieves the best results, followed by the contextual modeling method, and finally the contextual pre-filtering method; (c) More importantly, contextual methods perform better than non-contextual methods, meaning that including contextual information into CF improves the recommendation quality.

The overall solution can be implemented in LBS to provide tourists with personalized and context-aware location recommendations when visiting a new environment (e.g., city or museum). As our approaches do not require an explicit representation of domain knowledge, they are very suitable for LBS, which might often need to provide services in scenarios with little (or no) available domain knowledge. Additionally, our approaches employ a non-intrusive user modeling technique and do not require users to state their preferences explicitly, which are very promising in LBS, as LBS users are often involved in many tasks and activities during their use of mobile devices. Furthermore, our approaches can provide users with personalized and context-aware recommendations, which are very welcome in LBS, as context-awareness plays a key role in LBS applications.

The insights gained in this research can be transferred to many other applications, such as friend recommendations in location-based social networks, artwork recommendations in museums, recommendations in the shopping domain, human behavior understanding, and activity recognition.

Acknowledgements

I am profoundly grateful to Prof. Dr. Georg Gartner for providing me with the opportunity to undertake my PhD project within his research group. He introduced me to the field of Location Based Services, and kept me motivated and guided through my PhD studies. I am grateful for the time he devoted to discussing with me. It is a big pleasure and benefit to work with such a strong scientific advisor and a friend with a great personality.

I also express my gratitude to my second supervisor Prof. Dr. Martin Raubal from ETH Zürich for providing advice and support when needed, and his insightful and constructive comments when reviewing drafts of my papers and thesis. I also want to thank him for hosting me when I visited his group at The University of California, Santa Barbara (UCSB). It was really a very nice experience there.

I am also grateful to my colleagues Dipl.-Ing (FH) Manuela Schmidt, Dipl.-Ing Felix Ortag and Mag. BA Silvia Klettner whom I closely work with. I really enjoyed discussing with them. I also want to thank them for encouraging me to have vegetarian lunch, and for helping me to solve problems whenever German language is needed. I want to express my thanks to Ms. Violet Derman (Ibi) who is our secretary. I still remember that when I arrived at Vienna for the first time, Ibi and her husband picked me up at the airport and drove me to my hostel. Ibi has been always very helpful when I have some administrative issues. I also want to thank my other colleagues Prof. Mirjanka Lechthaler, Dr. Eszter Simonné-Dombóvári, Dr. Markus Jobst, Edith Wandl, Dipl.-Ing. Alexandra Millonig and Prof. Günther Retscher for being always helpful and kind to me. I feel very lucky to work in our family-like office.

I also want to thank the Austrian Science Fund (FWF), Österreichische Forschungsförderungsgesellschaft (FFG), and the Austrian Marshall Plan Foundation for supporting my research in Vienna. I want to express my thanks to the students from the course LBS 2011 for helping me collect the Vienna zoo dataset, to Prof. dr. ir. S.C. van der Spek from Delft University of Technology for sharing the Delft city dataset, to Connie Kwok for sharing the Flickr dataset, and to Prof. Dr. Nico Van de Weghe for his constructive comments.

I also want to forward my thanks to my colleagues in China, Prof. Yunpeng Wang, Prof. Yan Li, Prof. Guobin Chi, and Prof. Yang Cao, who supported me when I was in need of help.

Special thanks go to my mother, my father, my sister, my wife and my kid for their endless love, understanding, and unconditional support throughout my life. I love them.

Contents

List of Figures

List of Tables

Acronyms

AR	Augmented Reality
CaCF	Context-aware Collaborative Filtering
CF	Collaborative Filtering
DM	Domain Model
GPS	Global Positioning System
GSM	Global System for Mobile Communications
LBS	Location Based Services
LCS	Longest Common Subsequence
IDF	Inverse Document Frequency
POI	Point of Interest
RFID	Radio-frequency identification
RS	Recommendation system
SMoT	Stops and Moves of Trajectories
TF-IDF	Term Frequency-Inverse Document Frequency
UI	User interface

1 Introduction

1.1 Motivation

Recent years have witnessed rapid advances in Location Based Services (LBS) with the continuous evolvement of mobile devices and communication technologies. At a high level of abstraction, LBS are computer applications that deliver information depending on the location of the device and user (Raper *et al.* 2007a). Recently, LBS have become more popular in not only citywide outdoor environments, but also shopping malls, museums, and many other indoor environments. They have been applied in emergency services, tourism services, intelligent transport services, gaming, assistive services, etc. (Raper *et al.* 2007b). Among them, mobile guides (LBS for tourists, such as city guides and museum guides) are the largest group of LBS applications.

When visiting a new city, tourists often need help to effectively identify personally interesting places/locations from a potentially overwhelming set of choices. The task is further complicated by the physical attributes of the environment (Bohnert 2010), as it takes time for people to move between places, and personally interesting places may be scattered throughout the environment. LBS have a high potential to help tourists to solve this problem by providing location recommendations relevant to the current position, context, interests and needs of the users. This research investigates methods for making personalized and context-aware location recommendations in LBS.

Recently, with the increasing availability of GPS (Global Positioning System)-enabled devices, more and more people start to record their travel experiences with GPS loggers, and then upload, visualize and browse their GPS data on web maps. A large number of GPS data sharing websites have appeared on the Web, and received considerable attention, for example, Bikely[1], Wikiloc[2], GPS Share[3]

---

[1] http://www.bikely.com/

[2] http://www.wikiloc.com/

[3] http://www.gpsshare.com/

and Trimble Outdoors[4], just to name a few. Therefore, large amounts of location histories (e.g., GPS trajectories) are potentially available. In the meantime, with the rapid advances in geotagged[5] social media, recent years also witness many people publishing their travel information and experiences via social media, such as Foursquare[6] check-ins and Flickr[7] photos. These "self-reported" information and implicit "footprints" can be also used to construct users' location histories. Research on using these location histories often focuses on developing data mining methods to detect significant locations of a user (Liao *et al.* 2005, Ye *et al.* 2009), infer transportation modes (Zheng, Chen, *et al.* 2010), and analyze people's behavior patterns (Giannotti *et al.* 2007, Han *et al.* 2008, Nanni 2013, Renso and Trasarti 2013). However, these location histories also reflect people's travel experiences in the environment. Research has shown that experiences from past users (especially similar users) in similar contexts can help the current users efficiently to solve their problems (Wexelblat 1999, Zheng, Xie, *et al.* 2010). Therefore, by aggregating the experiences (as recorded in location histories) from other people, LBS can provide the current user with "social advice" for making decisions, such as services like "in similar contexts, other people similar to you often …".

This research aims to explore methods for deriving personalized and context-aware location recommendations in LBS. We employ the recommendation technique of Collaborative filtering (CF), which recommends a user the items that other users with similar tastes liked/used in the past. CF is often applied in Web-based applications, such as movie recommendations, music recommendations and product recommendations (Hayes and Cunningham 2004, Adomavicius *et al.* 2005, Karatzoglou *et al.* 2010, Panniello and Gorgoglione 2012). In this research, CF is enriched with contextual information, and integrated into LBS to aggregate similar users' movements (experiences) for making personalized and context-aware location recommendations for the current user.

Our research differs from other research on personalized and context-aware LBS services on the following aspects.

1) There was a large body of research on location-aware and personalized services (Oppermann

---

[4] http://www.trimbleoutdoors.com/

[5] Geotagging is the process of adding geographical location information (e.g., latitude/longitude coordinates) to various media, such as videos, photos and texts (Luo *et al.* 2011).

[6] http://foursquare.com

[7] http://www.flickr.com/

and Specht 1999, Sparacino 2002, Reichenbacher 2004, Wakkary and Evernden 2005, Sarjakoski and Sarjakoski 2005, Cena *et al.* 2006, Stock and Zancanaro 2007, Weakliam *et al.* 2008, Raubal and Panov 2009, Kenteris *et al.* 2011, Kim and Park 2011, Chen *et al.* 2013). Most of the research focused on adapting content presentation and visualization to users' current position, tasks, context, personal interests, and devices. In contrast to the above research, we aim to help people identify locations matching their interests and contexts, by deriving context-aware location recommendations from location histories.

2) Many approaches proposed for generating recommendations employed knowledge about locations/places (domain model, DM), knowledge about users and their context (user model), and an adaptation engine (Cheverst *et al.* 2000, Schmidt-Belz *et al.* 2002, Hagen *et al.* 2005, Park *et al.* 2007, Bader *et al.* 2011, Yılmaz and Erdur 2012, Yu and Chang 2013). The engine measured the appropriateness of a specific location for satisfying a particular user's interests, need and context, and returned relevant objects. However, building DM and the adaptation engine requires a good understanding of the application domain and has to undergo a long process of knowledge acquisition, both of which are very time-consuming and impractical for many LBS applications. Most importantly, these approaches were unable to effectively provide users with personalized and context-aware recommendations in situations with little (or no) previous knowledge, which are very common in LBS applications. In contrast to the above approaches, we aim at using CF to provide context-aware recommendations. CF requires little domain knowledge about the recommendation scenarios, as it generates recommendations solely based on users' feedback (e.g., explicit or implicit ratings) on items. In this sense, our approaches are very suitable for LBS applications, which might often need to provide services in scenarios with little (or no) available domain knowledge.

3) There were also LBS providing recommendations by asking users to explicitly state their interests in some form (Cheverst *et al.* 2000, Hagen *et al.* 2005, Horozov *et al.* 2006, Park *et al.* 2007, Li *et al.* 2009, Hung *et al.* 2012, Yang and Hwang 2013). In contrast to these systems, we investigate techniques of deriving context-aware location recommendations from non-intrusive observations of users (i.e., location histories), which do not require users' explicit inputs. These kinds of techniques are very promising and preferred in LBS applications due to the following reasons. Firstly, explicit inputs bring some burden to users, and interrupt

normal patterns of users' action (Nichols 1997). Secondly, LBS users are often involved in many tasks and activities during using mobile devices, which makes it difficult to state their interests explicitly. This is further complicated by the small size of mobile screens and restricted input functionalities. A field study conducted by Filippini Fantoni (2003) showed that 94% of the participants preferred systems that change their services automatically according to the users' need. Filippini Fantoni (2003, p. 4) argued that what makes this technique so exceptional is that "little effort is required on the part of the user because individual profiles are automatically built based on normal use."

4) There was also research focusing on deriving personalized recommendations from non-intrusive observations of users, such as location histories or interaction histories (Han and Cho 2006, Takeuchi and Sugimoto 2006, Bohnert 2010, Zheng *et al.* 2011, 2012, Yoon *et al.* 2012, Cheng *et al.* 2013). For example, Takeuchi and Sugimoto (2006) recommended shops to users based on their preferences, estimated by their past GPS trajectories. Han and Cho (2006) combined self-organizing maps and Markov models to predict users' movements based on past GPS trajectories. Bohnert (2010) analyzed users' location histories to extract their duration at exhibits and frequency counts of transitions between exhibits, and aggregated the extracted information to provide exhibit recommendations in a museum. Zheng *et al.* (2011) modeled users' GPS trajectories as hierarchical graphs, and made place recommendations by considering the sequence property of users' movement and the hierarchy property of geographic spaces. Our research differs from the methods mentioned above mainly on three aspects. Firstly, our methods are developed based on the concepts of stops and moves, which provide a fundamental and common framework for processing different kinds of location histories, such as GPS trajectories and trajectories constructed from Foursquare check-ins. In this sense, our methods are not restricted to a specific kind of location history. Secondly, we investigate whether considering the order in which locations/places are visited, location popularity, duration at locations and transit time between locations contributes to the improvement of recommendation quality. More importantly, most of the other research only employed location as contextual factor, and did not consider other contextual factors that might be potentially relevant for generating recommendations, e.g., weather, companion (with whom), and weekend/weekday. Our

research addresses these issues, and explores a methodology of deriving context-aware locations recommendations from different types of location histories.

1.2 Challenges

Deriving context-aware location recommendation from location histories is challenging for the following reasons.

1) Building user profiles from feedback/opinions on items made over time is the first step when making CF recommendations. In most of the existing CF applications, such as movie recommendations, a user profile is often represented as a set of ratings given by the user on different items, and each rating is modeled as a triple <user, item, rating>. When making location/place recommendations based on location histories, the user profile model employed in traditional CF methods (i.e., rating-based) is insufficient, as it cannot model users' behavior of visiting various locations, e.g., in which orders these locations are visited, and transit time between locations. Furthermore, methods of extracting meaningful user profiles from raw location histories (e.g., GPS trajectories) should be developed.
2) Existing CF methods often work with rating-based user profiles. However, when deriving location recommendations from location histories, it is still unclear how users' interests in various locations and motion behavior of visiting such locations (as extracted from location histories) can be used to identify other users who are similar to the current one. Furthermore, the question of how these similar users' movements can be aggregated for generating recommendations is still a subject of research. Existing CF methods cannot be directly applied to these questions.
3) Existing methods on deriving recommendations from location histories fail to consider contextual information, such as weather and companion (i.e., with whom)[8]. For incorporating more contextual information into the recommendation algorithms, several challenges still exist. Firstly, in order to provide context-aware recommendations, user profiles should be annotated with

[8] "Weather" and "companion" are examples of contextual dimensions. However, it does not mean that they are always relevant for generating recommendations in LBS. We argue that whether they are relevant or not should be carefully evaluated. In Section 5.2, we propose a methodology of identifying contextual dimensions/parameters that are relevant for a specific recommendation task.

contextual information. A context (situation) can be characterized by a set of context parameters/dimensions, such as “weather”. Not all context parameters are relevant for generating recommendations. However, methods on identifying relevant context parameters for LBS applications are still missing[9]. Secondly, how contextual information can be integrated into CF for making location recommendations is still unclear. It is also unclear about whether considering contextual information can help to improve recommendation quality.

1.3 Research questions and contributions

Research aims. This research aims to answer the question of how human location histories can be used to derive location recommendations in LBS. Specifically, we are interested in *how human location histories (e.g., GPS trajectories and trajectories constructed from people’s “self-reported” information on social media), and collaborative filtering (CF) can be integrated in LBS to provide users with personalized and context-aware location recommendations*.

In this research, the locations for recommendation are places that one may find useful or interesting. Similar to Zheng *et al.* (2011) and Li *et al.* (2008), we understand these kinds of locations as geographic regions, each of which can be represented as a topologically closed polygon in the geographic space. A location or geographic region may contain several points of interest (POIs), such as shops and restaurants[10]. In the following, we use location, place, and geographic region interchangeably.

---

[9] One exception was given by Keßler (2010), who developed a cognitively plausible dissimilarity measure to compare information retrieval result rankings (DIR). DIR was employed to identify which context parameters heavily influence the outcome of the retrieval task, and should therefore be modeled as relevant ones. This approach required that the ranking result for each context is available in advance, which might be hard to generate. In Keßler’s Surf Spot Finder, he used the domain knowledge about each spot and rules to generate ranking result for each context. However, as mentioned before, this knowledge-based approach requires a good understanding of the application domain and has to undergo a long process of knowledge acquisition (for building rules), both of which are very time-consuming and impractical for many LBS applications.

[10] Due to the constraints of current location-acquisition technologies, it is not always possible to identify the exact POI users are visiting, especially in a dense urban environment. Therefore, we make location recommendations instead of POI recommendations. However, we argue that with more accurate location-acquisition technologies, POI recommendation can be achieved by employing the methods proposed in this research.

The overall research question is comprehensively addressed in the following sub-questions:

Sub-Question 1: *How can a user's interests in various locations/places and motion behavior of visiting such locations, which are required for CF, be modeled and extracted from his/her location histories?*

Building a profile from the feedback of a user is the first step of CF. Existing CF approaches often represent a user profile as a set of ratings given by the user on different items. These kinds of user profile models are insufficient for making location recommendations from location histories, as it cannot model users' behavior of visiting various locations, e.g., in which orders these locations are visited, and transit time between locations. In this research, we first explore a model of contextual user profiles based on the concepts of stops and moves, which were introduced by Spaccapietra *et al.* (2008) and have been shown to be a useful abstraction of raw location histories (e.g., raw GPS trajectories) in the literature (Palma *et al.* 2008, Bogorny *et al.* 2009, Andrienko *et al.* 2011, Yan *et al.* 2011, Renso *et al.* 2013, Rinzivillo *et al.* 2013). With the proposed model, users' interests in different locations as well as their behavior of visiting such locations can be represented. In order to extract meaningful user profiles from raw location histories (e.g., raw GPS trajectories), a duration-threshold-free SMoT (DTF-SMoT) and a stay-point-based SMoT (SP-SMoT) are developed.

Sub-Question 2: *How can other users' interests in various locations and motion behavior of visiting such locations, as extracted from their location histories, be utilized to provide the current user with personalized location recommendations?*

Existing CF methods often work with rating-based user profiles, and are not suitable for deriving location recommendations from location histories. This research investigates how users' interests in various locations (reflected by their visits to these locations and the duration of these visits[11]) and motion behavior of visiting such locations can be combined to identify other users who are similar to the current user. More specifically, we explore a novel user similarity measure by considering the sequence property of movement (i.e., the order in which various locations are visited), location popularity, duration at locations and transit time between locations. We then

[11] This is in line with Bohnert (2010) and Zheng *et al.* (2011), which also used a user's stop and the stop duration at a location to approximate his/her implicit interest rating for the location.

employ this user similarity measure to identify users who are similar to the current one, and aggregate the “opinions” (i.e., movements) of these similar users to generate location recommendations for him/her.

Furthermore, we empirically investigate whether considering the order in which locations are visited, location popularity, duration at locations and transit time between locations contributes to the improvement of recommendation quality.

Sub-Question 3: *How can context-awareness be introduced to improve location/place recommendation in LBS?*

Existing methods on deriving recommendations from location histories often only consider users' current position and preferences, and fail to consider contextual information, such as weather and weekend/weekday, which may be potentially relevant for the recommendation tasks. In answering the former Sub-Question, we develop a non-contextual collaborative filtering (CF) method for deriving personalized location recommendations from a large number of users’ location histories. This method is further improved by integrating contextual information. Specifically, we develop a methodology for identifying context parameters that are relevant to the recommendation task, and explore a statistics-based approach for measuring similarity between different contexts (situations). We then design three approaches to integrate the similarity measure into CF for making context-aware location recommendations.

The methods developed for answering the above questions are comprehensively evaluated with three real-world datasets: 1) contextual GPS dataset in Delft city center (The Netherlands); 2) contextual GPS dataset collected from Vienna zoo (Tiergarten Schönbrunn, Austria), and 3) trajectories constructed from Flickr photos uploaded for the city of Vienna. These datasets consist of different types of location histories and reflect different scales of application scenarios.

1.4 Organization of the dissertation

This dissertation is structured as follows:

Chapter 1 (Introduction) introduces the background, motivation, and research aims, and outlines the contributions.

Chapter 2 (State of the Art) provides a state-of-the-art survey of related research, such as LBS, recommender systems, and applications of location histories.

Chapter 3 (Implicit User Profiling from Location Histories) addresses the Sub-Question 1, and mainly focuses on exploring a model of contextual user profiles, and methods of extracting meaningful user profiles from raw location histories.

Chapter 4 (Personalized Location Recommendations from Location Histories) addresses the Sub-Question 2, and mainly investigates how other users' interests in various locations and motion behavior of visiting such locations, as extracted from their location histories (as in Chapter 3), can be utilized to provide the current user with personalized location recommendations. Please note that the CF method proposed in this chapter does not use contextual information like weather and companion (with whom). Therefore, it can be considered as a non-contextual CF model.

Chapter 5 (Improving Location Recommendations through Context-awareness) addresses the Sub-Question 3, and aims to improve the non-contextual method (developed in Chapter 4) by integrating contextual information like weather and companion (e.g., alone or with others).

Chapter 6 (Conclusions and Future Work) concludes the research by summarizing the main contributions of this dissertation, and points out the future research directions.

2 State of the Art

Our research concerns how human location histories (e.g., GPS trajectories) and collaborative filtering (CF) can be integrated in LBS to provide users with personalized and context-aware location recommendations. This chapter provides a brief discussion of related research. Section 2.1 discusses the field of LBS and mobile guides. In Section 2.2, we discuss different recommendation techniques, and focus on recent advances in CF. Section 2.3 discusses research on using human location histories. Section 2.4 provides a brief survey of related LBS systems.

2.1 LBS and mobile guides

At a high level of abstraction, LBS are computer applications that deliver information/services depending on the position of device and user (Raper *et al.* 2007a). Recently, LBS have become more popular in not only citywide outdoor environments but also shopping malls, museums and many other indoor environments. They have been applied in emergency services, tourism services, intelligent transport services, gaming, assistive services, etc (Raper *et al.* 2007b). Among them, mobile guides (LBS for tourists, such as city guides or museum guides) are the largest group of LBS applications.

LBS often consist of three basic modules (Huang and Gartner 2010): positioning, modeling and adaptation, and information presentation and user interface (UI). The positioning module determines the current location of a user. For outdoor LBS applications, GPS is often employed. For indoor applications, additional installations (e.g. WiFi, Bluetooth and Radio-frequency identification RFID) are required (Retscher 2007). How to provide reliable and stable positioning information in complex and changing environments (indoor or outdoor) is still a challenging research question. Modeling and adaptation aim to model the user and his/her context, and intelligently adapt the services to them.

Research on information presentation and UI focuses on exploring technologies to convey/communicate information efficiently to the user, such as Augmented Reality (AR) and mobile maps (Gartner 2013). In many LBS applications, the last two modules may interconnect with each other.

2.1.1 Context-awareness and adaptation

For effectively supporting users, LBS should provide information and services adapted to the current location, context, and need of a mobile user. From this sense, context-awareness and adaptation play an essential role in LBS (Raper *et al.* 2007a).

The term context-aware computing was first introduced by Schilit *et al.* (1994). Since then, numbers of definitions of the term context have been proposed in the literature. Among them, Dey proposed a broadly adopted definition of context in computer science: "Context is any information that can be used to characterize the situation of an entity. An entity is a person, place or object that is considered relevant to the interaction between a user and an application" (Dey 2001, p. 5). Dourish (2004) further studied the definition of context, and identified two views for context: representational and interactional. The representational view understood that "context is a form of information, it is delineable, it is stable, and it is independent from the underlying activity" (Adomavicius and Tuzhilin 2011, p. 68). This view assumed that context acts as a set of conditions under which an activity (i.e., interaction) occurs, and it could be modeled using a set of observable attributes of these situations. These attributes can be considered as context parameters, which can help to differentiate/recognize different situations. These parameters were known at the stage of system development, and the structure of them did not change over time. On the opposite, in the interactional view, the scope of context was defined dynamically; therefore, enumeration of context conditions was not possible beforehand. This view assumed "a cyclical relationship between context and activity, where the activity gives rise to context and the context influences activity" (Adomavicius and Tuzhilin 2011, p. 68). Compared to the interactional view, the representational view is much simpler and more computationally feasible for many applications (Baltrunas 2011). Therefore, similar to the majority of relevant work, the representational view is also adopted in this dissertation.

When developing context-aware systems the developer must pre-determine what aspects of the world can be considered as context parameters, i.e., the attributes characterizing the situations that the system might encounter. Different categories have been proposed for context. Dey (2001) categorized context into primary context and secondary context. Primary context elements were place, time, identity, and activity. The primary context elements can work as indices to secondary elements, e.g., weather conditions. Schmidt *et al.* (1999) distinguished context related to human factors and context related to the physical environment. The former one was structured into three categories: information on the user, the user's social environment, and the user's tasks. Context related to the physical environment was structured into three categories: location, infrastructure, and physical conditions. Nivala and Sarjakoski (2003) developed a classification of context for map-based mobile services: user, location, time, orientation, navigation history, purpose of use, social and cultural situation, physical surroundings, and system. Reichenbacher (2004) structured context into different dimensions in the field of mobile cartography: situation, user, activities, information, and system. The described classifications provide some structures for consideration of context (Schmidt *et al.* 1999). For practical uses of context, the general challenge is to identify which attributes (i.e., context parameters) are relevant and needed to be modeled for context-aware services. Current context-aware systems often choose some features as context parameters from their own views (Huang and Gartner 2009). What is missing, however, is a method about how to identify relevant context parameters. We address this issue when discussing the context-aware recommendation algorithms (Chapter 5).

Recognizing the context (situation) that the user is currently in is another key issue in context-aware services. As mentioned above, a situation can be characterized as a set of relevant parameters. The values of these parameters can be provided either by users themselves or by different sensors. Different approaches have been proposed for using sensor output for situation recognition (Schmidt 2002, Coutaz *et al.* 2005, Ye 2009). Coutaz *et al.* (2005) proposed three layers of abstraction for translating sensor output into situation recognition: the sensor layer, the perception layer, and the situation/context identification layer. The sensor layer collected numeric output of a collection of sensors. Some transformations were needed to determine meaning from numeric observables. The perception layer was independent of the sensing technology and provided symbolic observables at the appropriate level of abstraction. The situation/context identification layer identified the current situation and context from symbolic observables, e.g., by reasoning. Ye (2009) proposed a lattice theory based approach to infer situations from sensor output. Each type of sensor reading was considered as a context predicate. A situation's specification can be expressed as a logical description that takes context predicates as input and applies the logical operators on them. The specifications of different situations can be learnt from training data.

When the context/situation of the current user is identified, adaptation techniques can be used for fitting the services according to the current context (Reichenbacher 2004, Raubal and Panov 2009). Fischer (1993) proposed two ways for achieving adaptation: adaptable and adaptive. This classification was a differentiation between manually and automatically performed adaptation processes. Adaptable systems enable users to customize the functionality of the services by themselves. The advantage of this approach is that "the user is in control", while the disadvantage is that "the user must do substantial work". On the contrary, adaptive systems adapt their services automatically according to the user and his/her context. The advantage of this approach is "little (or no) effort by the user", while it has the weakness of "loss of control". Raubal and Panov (2009) argued that services requiring large amounts of explicit interaction have less potential of being used, as they tend to be obstructive. A field study conducted by Filippini Fantoni (2003) showed that 94% of the participants preferred systems that change their services automatically according to the users' need. Therefore, considering the fact that LBS users are often involved in many tasks and activities during using mobile devices, adaptive services rather than adaptable ones should be introduced into LBS.

This dissertation aims to provide LBS users with adaptive location recommendations matching their interests and context, such as weather and companion (with whom). Location/place recommendations are derived from users' location histories, which can be considered as non-intrusive observations of these users' previous movements. With this, we expect to provide users with context-aware location recommendations while requiring little (no) explicit interaction from them.

2.1.2 Mobile guides and location recommendations

Mobile guides are the largest group of LBS applications (Raper *et al.* 2007b). In this section, we survey related research on mobile guides, and focus on location recommendations.

Mobile guides aim to provide users with enjoyable and informative guidance when visiting an unfamiliar environment, such as a new city or a new museum. When travelling around a new environment, tourists often need help to identify personally interesting places/locations from a potentially overwhelming set of choices. The task is further complicated by the physical attributes of the environment (Bohnert 2010), as it takes time for people to move between places, and personally interesting places may be scattered throughout the environment. Furthermore, when visiting a place, tourists often expect information about the place, such as history, stories, and other relevant information. In other words, they need personalized information about the places they visit.

For providing place/location recommendations and information about places in mobile guides, personalization and context-awareness play an essential role. There is a large body of research focusing on providing users with personalized information about places, and location-tailored and context-tailored visualization (Oppermann and Specht 1999, Sparacino 2002, Sarjakoski and Sarjakoski 2005, Wakkary and Evernden 2005, Cena *et al.* 2006, Stock and Zancanaro 2007, Tallon and Walker 2008, Kenteris *et al.* 2011, Kim and Park 2011, Chen *et al.* 2013). For example, Oppermann and Specht (1999) developed a museum guide, in which content selection and presentation of exhibits were adapted to the current device, network connection, current location, as well as users' knowledge and preferences. Sarjakoski and Sarjakoski (2005) visualized relevant POIs on map views, while the visualization was adapted to different seasons. Cena *et al.* (2006) adapted the content of the mobile guide being provided and presentation according to the device, user preferences, and the context (i.e., the user location and time of the day). In summary, the research focused on providing and presenting information adapting to users' interests, tasks, and context.

There are also mobile guides aiming to provide location recommendations matching users' interests, need, and context. For example, Cheverst *et al.* (2000) recommended attractions according to the current location, users' interests, and the opening time of the attractions. A static user model obtained from explicit user input was employed to generate a tailored city tour. Hagen *et al.* (2005) employed ontology to identify locations relevant to users' interests, the available time period, and the current position. Users need to explicitly state their interests. Horozov *et al.* (2006) developed a restaurant recommendation system for mobile users. Restaurants were recommended based on the current location and users' interests. It proposed a location-enhanced collaborative filtering (CF) method, which required a large amount of explicit ratings about nearby restaurants. Li *et al.* (2009) proposed a multi-stage collaborative filtering method to provide event recommendations based on the current location and users' interests. However, the approach required users to provide their profiles explicitly. While these approaches successfully provided users with relevant location recommendations, they had the drawbacks that users were required to state their interests explicitly. As LBS users are often involved in many tasks and activities during using their devices, mobile guides employing the above approaches might tend to be obstructive, and therefore have less potential of being used (Filippini Fantoni 2003, Raubal and Panov 2009).

In recent years, many researchers have started to explore methods that provide location recommendations to users in a non-intrusive manner, by learning user profiles from their interaction histories or location histories. For example, Schmidt-Belz *et al.* (2002) developed a mobile guide adapted to the current location and user interests. It learnt a user's interests from his/her interactions with the system. Services were then tailored to user interests by using a domain taxonomy. van Setten *et al.* (2004) reported on the COMPASS project, which provided location-based personalized POI recommendations to users. A user's interest model was manually initialized and further automatically updated by the system based on the user's feedback for specific POIs. An ontology describing the class hierarchy of POIs was employed during the recommendation process. Takeuchi and Sugimoto (2006) recommended shops to mobile users by considering the current location and users' interests. Users' interests were estimated by analyzing their past location history recorded by GPS. An item-based CF was employed for recommending nearby shops matching users' interests. Bohnert (2010) analyzed users' location histories to extract their interests, and aggregated the extracted information to provide exhibit recommendations in a museum. Zheng *et al.* (2011) made friend recommendations and place recommendations by mining a large amount of GPS trajectories. The place recommendations were mainly generated by employing CF technique.

Several drawbacks of the above non-intrusive methods should be mentioned. Firstly, many of the non-intrusive methods employed knowledge bases (e.g., adaptation rules) or ontologies for making location recommendations (e.g., Schmidt-Belz *et al.* 2002, van Setten *et al.* 2004). For building the knowledge bases and ontologies, a long underlying learning (knowledge acquisition) process has to be carried out, and knowledge about the application domain should be extracted, both of which are very time-consuming and impractical for lots of LBS applications. More importantly, these approaches were not able to provide users with context-aware services in situations with little available (or no) domain knowledge, which are very common in LBS applications. Secondly, for other systems employing a collaborative filtering technique rather than knowledge-based technique, most of them only provided location/place recommendations according to users' interests and current position, and did not consider other contextual factors that may be potentially relevant for generating recommendations, e.g., weather, companion (with whom), and weekend/weekday.

Summary. Mobile guides have gained increasing interest in recent years (Stock and Zancanaro 2007, Wiesenhofer *et al.* 2007, Yılmaz and Erdur 2012, Cheng *et al.* 2013, Yang and Hwang 2013). Different approaches have been applied for providing personalized and context-aware services in mobile guides. However, many of them focused on adapting content presentation rather than location recommendations. There were also many mobile guides generating location recommendations by asking users to state their interests explicitly. As LBS users are often involved in many tasks and activities during using mobile guides, these intrusive recommendation approaches might have less potential of being used. Furthermore, some of the mobile guides employed a knowledge-based approach for adaptation. These knowledge-based approaches often required a long underlying learning (knowledge acquisition) process and a good understanding of the application domain, both of which are very time-consuming and impractical for many LBS. There were also systems employing the collaborative filtering technique, and avoiding the problem of "knowledge acquisition bottleneck". However, they often generated recommendations only according to users' interests and current location.

This dissertation addresses the above challenges. We mainly focus on deriving context-aware location recommendations from location histories (such as GPS trajectories), which can be considered as non-intrusive observations of users' previous movements. Collaborative filtering (CF) is employed in the recommendation process. In the following, related work on recommendation systems and mining location histories is summarized.

2.2 Recommendation systems

Recommendation systems (RSs) aim to provide a user with a personalized list of items that are relevant to his/her interests, need, and context. Various types of information about users, items, and interactions between users and items are often collected and exploited for making recommendations. In terms of interactions between users and items, the most commonly used information is the set of subjective ratings assigned by the users to previously experienced items. The system then uses these ratings to predict the ratings for items not yet experienced. Items with higher estimated ratings will be recommended to the user (Baltrunas 2011). RSs have been applied in many domains, such as movies, music, news, jokes, city tours, museum, and e-commerce (Bohnert 2010).

2.2.1 RS techniques

A variety of techniques has been proposed for RSs. Among them, collaborative filtering, content-based filtering, knowledge-based filtering, and hybrid RS are the most popular ones (Hanani *et al.* 2001, Ricci *et al.* 2011).

Collaborative filtering (CF) provides a user with the items that other users with similar tastes liked in the past (Resnick and Varian 1997). The recommendations on the amazon.com website ("people who bought … also bought …") are well-known CF examples. CF is a domain independent approach, which makes recommendations based on users' opinions/ratings on different items. A rating is often modeled as a tuple <user, item, rating>. Ratings can be expressed explicitly, e.g., by indicating a rating on a scale, or can be inferred implicitly, e.g., from purchase behavior or moving trajectories. User-based CF is a typical CF approach. It identifies users that are similar to the current users, and aggregates these users' "opinions" to generate recommendations for the current users.

The biggest advantage of CF is that it requires little domain knowledge about the recommendation scenarios, as it generates recommendations solely based on ratings. CF can also recommend items outside the observed interests of a user (i.e., "surprising" items), and this is possible because recommendations are based on the ratings of other similar users. However, pure CF has some disadvantages (Desrosiers and Karypis 2011). Two of them are data sparsity (too few common ratings) and cold-start problem (new user problem and new item problem). For example, pure CF cannot generate recommendations for new users, as no ratings from them are available.

Content-based filtering (CBF, or content-based recommendation) recommends items (e.g., locations) similar to those the user has liked in the past. Only the ratings of the current user are exploited for making recommendations. These systems build a model or profile of user's preferences based on the features (description) of the objects rated/chosen/liked by that user (Lops *et al.* 2011). A profile is a structured representation of a user's preferences. The recommendation process matches up the user profile with the attributes of an item (item profile). The result is a relevant judgment representing the user's level of interest in that item. Items with higher relevant judgment values are often recommended to the end user. The performance of content-based RSs mainly depends on how accurate the profile reflects the user's preferences. Recently, approaches aiming to integrate context-awareness into CBF have been also proposed in the literature (Yap *et al.* 2005, 2007).

CBF systems do not suffer from the new item problem. Therefore, they are preferred to CF in those domains where the main need is to recommend recent items or data sparsity is very high (Pazzani and Billsus 2007), for example in the news domain. CBF systems also have several limitations (Adomavicius and Tuzhilin 2005, Lops *et al.* 2011), such as limited content analysis in building profiles (for both users and items), and overspecialization. The former one refers to the aspect that the recommendation quality of CBF depends on the availability of information about the items. If little description about items is available, it is hard to build profiles that accurately reflect the characteristics of items and preferences of users. Overspecialization means content-based RSs have no inherent method for finding something "surprising" (McNee *et al.* 2006, Lops *et al.* 2011).

Knowledge-based Filtering (KBF, or knowledge-based recommendation) recommends items based on *predefined* knowledge bases that contain explicit rules about how certain item features meet users' need and preferences, or ultimately, how useful the item is for the user (Schafer *et al.* 1999, Felfernig *et al.* 2011).

A knowledge base is typically defined by two sets of variables (V_U, V_{PROD}) and three different sets of constraints (C_R, C_F, C_{PROD}) (Felfernig *et al.* 2011). User Properties V_U describe possible requirements of users, i.e., requirements are instantiations of user properties, which may be explicitly provided by users via a series of dialogs. Product Properties V_{PROD} describe the properties of a given product assortment. Constraints C_R systematically restrict the possible instantiations of user properties. Filter Conditions C_F define the relationship (rule) between potential user requirements and the given product assortment. Products C_{PROD} store all the products, and represent them by using the properties defined in V_{PROD}. Among them, Filter Conditions C_F plays a key role. An example of a C_F rule can be

C_F = {C_{F1}: With_cash = not → Credit_cards_accepted = Yes}

It can be explained as "users without cash should receive recommendations (restaurants) that accept credit cards".

Compared to CF and CBF, KBF has no cold-start problems since users' requirements are directly elicited within a recommendation session through a series of dialogs. However, it suffers from "the knowledge acquisition bottleneck in the sense that knowledge engineers must work hard to convert the knowledge possessed by domain experts into formal, executable representations" (Felfernig *et al.* 2011, pp. 187–188). Therefore, it is often combined with other RS techniques.

Hybrid RS. As mentioned above, each RS technique has advantages and disadvantages. Hybrid RSs combine two or more of the above techniques. A hybrid system combining techniques A and B tries to use the advantages A to fix the disadvantages of B (Ricci *et al.* 2011). For instance, pure CF suffers from the cold-start problem (new item and new user), i.e., they cannot recommend items that have no ratings, and they cannot make recommendations to users who have not given ratings. These can be solved by applying a knowledge-based technique at the beginning. Adomavicius and Tuzhilin (2005) and Burke (2002) provided some surveys on hybrid RSs.

Ricci *et al.* (2011) distinguished RS into additional classes, i.e., demographic RS and community-based RS. Demographic RS recommends items according to the demographic user profile (age, language, country), while community-based RS recommends those items that the user's friends like.

In summary, different RS techniques have advantages and disadvantages, and require different inputs. Among these RS techniques, CF requires little domain knowledge about the recommendation scenarios, as it generates recommendations solely based on ratings. In terms of recommendations in LBS, we are aware that LBS (e.g., mobile guides) often need to effectively provide users with context-aware services in situations with little (or no) domain knowledge. Therefore, CF is a very promising technique for providing recommendations in LBS. Furthermore, with the increasing availability of GPS-enabled devices and geotagged social media, large location history datasets (e.g., GPS trajectories or trajectories constructed from social media) are potentially available for LBS. These location histories reflect people's travel experiences and implicit feedback about the environment, which enable CF with an abundance of data for accurate recommendation. To summarize, CF is a promising technique for making recommendations in LBS, especially location/place recommendations in mobile guides. Therefore, this research investigates how CF can be used in LBS to derive context-aware location recommendations from location histories. In the following, we provide a more detailed survey on collaborative filtering.

2.2.2 Collaborative filtering (CF)

As mentioned before, CF uses "opinions" of similar users to help the current user efficiently identify items of interest (Resnick and Varian 1997).

The first stage of CF is to build user profiles from users' feedback on items made over time. A user profile is often represented as a set of ratings given by the user on different items, and a rating is modeled as <user, item, rating>. Feedback can be explicit and implicit (Nichols 1997). Explicit feedback requires explicit actions from users (e.g., indicating a rating on a scale) which bring some burden to them, and interrupt normal patterns of their action (Nichols 1997). For implicitly collecting, the system tracks users' implicit feedback (i.e., moving tracks, interaction history) to unobtrusively infer their preferences. For example, Froehlich *et al.* (2006) found that there existed a positive correlation between explicit place ratings and implicit aspects of travel behavior such as visit frequency and travel time. It is also important to note that while rating-based approaches are effective for representing user profiles, especially in the domain of movie and product recommendations, they might be insufficient to represent the information extracted from users' movements. For example, when making location recommendations based on GPS trajectories, the rating-based approaches cannot effectively represent users' behavior of visiting various locations/places, e.g., in which orders these locations are visited, and transit time between locations. We address this issue in Chapter 3.

Algorithms for CF can be grouped into two general classes (Adomavicius and Tuzhilin 2005):

1) Model-based CF (Pavlov and Pennock 2002, Hofmann 2003, Marlin 2003, Karatzoglou *et al.* 2010) uses the collection of ratings to learn a model, which is then used to make rating predictions. Probabilistic models (such as Bayesian networks and cluster model) are often employed for model learning.
2) Heuristic-based or neighborhood-based CF (Resnick *et al.* 1994, Delgado and Ishii 1999, Desrosiers and Karypis 2011) can be divided into user-based approach and item-based approach. Given an unknown rating (of an item by the current user) to be estimated, neighborhood-based CF first measures similarities between the current user and other users (user-based), or between the item and other items (item-based). Then the unknown rating is predicted by averaging (weighted) the known ratings of the item by similar users (user-based), or the known ratings of similar items by the current user (item-based).

Research has found that state-of-the-art model-based approaches can achieve better prediction accuracy than neighborhood-based approaches (Takács *et al.* 2007, Koren 2008). However, neighborhood-based approaches also have their own advantages, such as simplicity, justifiability (the ability to provide a concise and intuitive explanation about why the items are recommended), efficiency, and stability (Desrosiers and Karypis 2011).

In recent years, researchers have started to investigate how CF can be improved by considering contextual information, such as shopping purposes, weather, and seasons. Several context-aware CF (CaCF) approaches have been proposed in the literature (Herlocker and Konstan 2001, Hayes and Cunningham 2004, Adomavicius *et al.* 2005, Oku *et al.* 2006, Anand and Mobasher 2007, Panniello *et al.* 2009, Karatzoglou *et al.* 2010, Baltrunas 2011, Panniello and Gorgoglione 2012). For example, Adomavicius *et al.* (2005) proposed a contextual pre-filtering CaCF method, in which ratings collected in other conditions as the current one were discarded, and a standard CF algorithm was then applied on this reduced set of data to generate contextual recommendations. Oku *et al.* (2006) incorporated context-aware Support Vector Machine (C-SVM) into CF for providing contextual recommendations. C-SVM was used to compute user similarities. Karatzoglou *et al.* (2010) extended Matrix Factorization (MF) by adding contextual information, and designed a Tensor Factorization (TF) to provide context-aware recommendations. To better understand different CaCF methods proposed in the literature, Adomavicius and Tuzhilin (2011) proposed to classify them into three groups, based on how (when) the contextual information is used. They were contextual pre-filtering (contextualization of recommendation input), contextual post-filtering (contextualization of recommendation output), and contextual modeling (contextualization of recommendation algorithms). Panniello and Gorgoglione (2012) empirically compared these three approaches for contextual product recommendations using transaction data (purchasing information) from e-commerce websites. They showed that there was no clear winner for all settings, while "filter" based post-filtering outperformed other contextual approaches in many settings. However, it is unclear whether this conclusion still holds for other application domains or not, as different domains may have different characteristics.

It is important to note that, most of the above CF methods and CaCF methods were designed for and evaluated in movie, music, and product domains, and employed explicit ratings. There were some attempts on applying CF in LBS for making recommendations, such as restaurant recommendations (Horozov *et al.* 2006), event recommendations (de Spindler *et al.* 2006, Li *et al.* 2009), shop recommendation (Takeuchi and Sugimoto 2006), exhibit recommendations in museums (Bohnert 2010), and place recommendations (Zheng *et al.* 2011). However, as mentioned in Section 2.1.2, many of these approaches required users to explicitly state their interests, e.g., Horozov *et al.* (2006), de Spindler *et al.* (2006) and Li *et al.* (2009). For other systems learning from users' behavior (Takeuchi and Sugimoto 2006, Bohnert 2010, Zheng *et al.* 2011), recommendations were often only adapted to users' interests and their current location. However, there are also many other contextual factors such as weather and companion, which might be relevant for generating recommendations.

This dissertation tries to address the above challenges. A non-intrusive context-aware CF method is developed to derive personalized and context-aware location recommendations from location histories, such as GPS trajectories and trajectories constructed from social media.

2.3 Mining location histories

With the availability of different tracking technologies (e.g., GPS), recent years have seen an increasing abundance of data about the trajectories of moving objects (such as cars, humans, animals, goods) being recorded. In the meantime, more and more trajectories can be constructed from users' "self-reported" information on social media, such as Foursquare check-ins, and Flickr photos. Such location histories play an essential role in a variety of novel applications in different scientific and social domains. Therefore, in recent years, research on movement data analysis and application has been blooming. Many research communities such as MODAP[12] (mobility, data mining, and privacy), MOVE[13], and SEEK[14] (SEmantic Enrichment of trajectory Knowledge discovery) are actively focusing on this issue.

A trajectory is defined as an evolution of the position (perceived as a point) of an object that is moving in space during a given time interval in order to achieve a given goal (Spaccapietra *et al.* 2008). It is often represented as a discrete sequence of points, with an interpolation function to approximate the movements between two consecutive points. Different technologies can be used for tracking trajectories of moving objects, such as GPS, GSM (Global System for Mobile Communications), Bluetooth, and RFID. As mentioned above, trajectories can be also constructed from social media.

Real-life trajectory data, collected using the technologies previously mentioned, are not ready for further applications. Such data are often imprecise due to limitations of tracking technologies (e.g., inaccurate GPS measure and sampling errors, signal loss). Therefore, trajectory data cleaning is often needed before the actual data analysis process (Marketos *et al.* 2013). Different filtering methods (e.g., filtering noisy points by considering maximum allowed speed of a moving object) and smooth methods (e.g., Gaussian kernel based local regression model, Kalman filtering) can be employed. For objects which are restricted to move within a given spatial network (e.g., road/railway network), map-matching approaches can also be used for data cleaning.

[12] http://www.modap.org/

[13] http://www.move-cost.info/

[14] http://www.seek-project.eu/

2.3.1 Semantic trajectory processing and mining

Early research mainly focused on the geometric properties of trajectories, and therefore tended to discover geometric patterns. Geometric patterns were normally extracted based on the concept of dense regions and trajectory similarity. However, without considering semantic information that is available from background geographic information, it is very difficult to have a meaningful interpretation of movement behavior based on the extracted geometric patterns (Yan *et al.* 2011). In order to extract meaningful knowledge, raw trajectories should be enriched and integrated with additional semantics, such as geographic information, events related to the movement, and activities performed by the moving object when it stopped (Spaccapietra *et al.* 2013).

Recently, adding semantics to trajectories has received increasing attention. Spaccapietra *et al.* (2008) proposed a conceptual view on trajectories, modeling a trajectory as a sequence of stops and moves. Stops were the important parts of a trajectory where the moving object did not move, as far as the application view of this trajectory is concerned. Moves were the sub-trajectories describing the movements between two consecutive stops. They were often based on the moving speed of the object, but the semantic expression depended on the application. Based on the concepts of stops and moves, users can enrich trajectories with semantic information according to the application domain. The conceptual view based on stops and moves has been applied in various trajectory-based applications, such as Palma *et al.* (2008), Bogorny *et al.* (2009), Yan *et al.* (2011), Andrienko *et al.* (2011), Renso *et al.* (2013), Rinzivillo *et al.* (2013), just to name a few.

Extracting stops and moves: Different approaches have been proposed to identify stops and moves from raw trajectories. Alvares *et al.* (2007) proposed the SMoT (Stops and Moves of Trajectories) method. The method required a set of pre-defined geographic places/areas (i.e., candidate stops) and their minimum duration thresholds as inputs. If an object had stayed in a pre-defined area for the duration longer than the duration threshold, it was considered to have stopped in this area. The method verified the intersection of the trajectory with this set of geographic areas and their duration thresholds to extract stops and moves. The main challenge of this method is the definition of a suitable set of geographic areas and their duration thresholds, which is application-dependent. The method CB-SMoT (Clustering-Based SMoT), proposed by Palma *et al.* (2008), is a clustering method based on the variation of the speed of the trajectory. It firstly extracted clusters (potential stops), where the speed was lower than a given threshold, for a minimal amount of time. In a second step, the method verified for each cluster if it intersected the pre-defined geographic areas. All clusters that intersected the geographic areas for a minimal amount of time were labeled with the names of the areas; otherwise, they were named as unknown stops. CB-SMoT can help to discover stops that are unknown *a priori*, but may be potentially interesting to the application. However, CB-SMoT is still very sensitive to the input parameters, such as the speed and thresholds. Rocha *et al.* (2010) proposed another clustering method DB-SMoT (Direction-Based SMoT), based on the variation of trajectory direction. Stops were generated for sub-trajectories where the direction variation is lower than a given threshold and for a minimal amount of time. This method is useful in specific domains where the direction variation has a greater impact than speed. Zheng *et al.* (2011) proposed a stay point based approach (tree-based hierarchical graph, TBHG) for processing raw GPS trajectories. A stay point stood for a geographic region where a user stayed over a certain time interval. For each trajectory, a set of stay points were extracted based on a time threshold and a distance threshold. As the stay points pertaining to different individuals were not identical, a density-based clustering algorithm was employed to hierarchically cluster all users' stay points into several geospatial regions in a divisive manner. The nearby stay points from various users were assigned to the same cluster on different layers. This organization of clusters provided various users with a uniform framework to represent their own location history (i.e., as a hierarchical graph). This approach was developed in the GeoLife project (Zheng, Xie, *et al.* 2010) for friend and location recommendations (see Section 2.3.2). It is important to note that, this approach was based solely on spatio-temporal properties, and returned some geometric clusters as output. It might be hard to annotate these clusters with semantics due to the absence of semantic information during clustering. To sum up, the above methods can help to extract stops and moves from raw trajectories. However, all the above methods are very sensitive to the input parameters. Defining suitable values for the input parameters is still a challenge.

Mobility data mining: A large body of research has focused on developing methods to understand moving objects' behavior based on their movement trajectories. In the following, we mainly focus on research using human movement data. There was research using personal movement to study individual behavior. Liao *et al.* (2005) detected personally significant places (e.g., home, workplace, shopping centers) and personally significant transportation routines (i.e., the paths and transportation modes) from GPS trajectories. Ye *et al.* (2009) proposed a mining approach using an individual's location history to extract significant-place-based life patterns, i.e., regularity of the ways of visiting these places. Many other studies have focused on *mining multiple users' location histories*. Giannotti *et al.* (2007) developed an extension of the sequential pattern mining paradigm that analyzed the trajectories of multiple moving objects, and extracted trajectory patterns which described frequent behavior of visiting the same sequence of places with similar travel times. In addition to sequential mining, other data mining techniques, such as clustering, classification and outlier detection, were often employed to discover interesting behavior patterns (Han *et al.* 2008, Yuan and Raubal 2012, Nanni 2013, Renso and Trasarti 2013). Visual analytics, which helps to interpret large amounts of movement data by interactive, visually-driven exploratory data analysis techniques, is another approach for extracting behavior patterns (Andrienko and Andrienko 2007).

2.3.2 Applications using trajectories

The increasing abundance of trajectory datasets has enabled many innovative applications in different scientific and social domains, such as ethology (Focardi and Cagnacci 2013), traffic management (Janssens *et al.* 2013), urban studies and planning (van der Spek *et al.* 2009), crowd management and safety at mass events (Versichele *et al.* 2012), marketing (Versichele *et al.* 2013), and Location-Based Services (e.g., the GeoLife project by Zheng, Xie, *et al.* (2010)). In the following, we mainly focus on LBS applications using trajectories.

The GeoLife project, implemented by Mircosoft Research Asia, aimed at understanding the correlation between users and locations in terms of user-generated GPS trajectories (Zheng, Xie, *et al.* 2010). In this project, the TBHG-based approach (please refer to the "*Extracting stops and moves"* part in Section 2.3.1 for an introduction) was employed for processing raw GPS trajectories. Different LBS applications have been developed in this project. Zheng *et al.* (2009) recommended classical travel sequences among locations (i.e., non-personalized), considering users' interests towards these locations and their travel experience levels. Users' interests towards a location and their travel experience were computed with a HITS (Hypertext Induced Topic Search)-based inference model. Zheng *et al.* (2011) proposed a CF-based approach to generate personalized place recommendations based on individual location history (GPS trajectory). Firstly, a hierarchical-graph-based similarity measure (HGSM), originally proposed by Li *et al.* (2008), was employed to uniformly model each individual's location history, and effectively measure the similarity among users. A set of unvisited locations that were liked (visited) by the current user's similar users were then recommended to him/her. Finally, a content-based method was also incorporated to alleviate the new item problem. Xiao *et al.* (2012) enriched the HGSM with semantic information for measuring user similarity. Similarity measure was then applied to friend recommendations. They showed that the new similarity measure outperformed the original one proposed in Li *et al.* (2008) and Zheng *et al.* (2011). In addition to friend and location recommendations, Yoon *et al.* (2012) proposed a social itinerary recommendation by learning from multiple GPS trajectories. Firstly, a Location-Interest Graph was learnt offline. This graph contained locations (interest, typical staying time) as vertices, travel time and classical level between two connected locations as edges. An online processing framework, composed of query verification, trip candidate selection, trip candidate ranking, and re-ranking by travel sequence, was employed to generate itinerary recommendations. It is important to note that, interest-based personalization has not been introduced into their approach. Zheng *et al.* (2012) proposed a model-based CF approach for location recommendations (e.g., "where should I go if I want to go shopping") and activity recommendations (e.g., "what can I do there if I visit this place"). Different information was extracted to feed the CF tasks: location features from POI database, activity-activity correlations from the Web, user-user similarities from the user demographics database, and user-location preferences from GPS trajectories. To sum up, in the GeoLife project, different approaches have been proposed to use GPS trajectories for novel applications, such as friend, location, and activity recommendations, which illustrate the benefits of using GPS trajectories in LBS. However, it is important to note that, contextual information, such as weather and companion information, has not been considered in the above approaches. Furthermore, their approaches were designed for GPS trajectories, and might be hard to apply for other trajectories, such as trajectories constructed from social media.

Other researchers have also proposed some innovative applications using trajectories. A significant number of articles have presented work aiming to mine GPS trajectories of taxi drivers for route recommendation for car navigation (Letchner *et al.* 2006, Gonzalez *et al.* 2007, Ziebart *et al.* 2008, Yuan *et al.* 2010). Many of them tried to provide routes with less travel time for car drivers. There were also applications mining trajectories for tourists. Takeuchi and Sugimoto (2006) recommended shops to users based on their individual preferences and need, estimated from their past location histories (GPS trajectories). Han and Cho (2006) combined self-organizing maps (SOM) and Markov models to predict users' movements based on past GPS trajectories. Bohnert (2010) mined visitors' moving tracks to provide exhibit recommendations in a museum. There is also research focusing on constructing travel itineraries based on geotagged photos on the Web (De Choudhury *et al.* 2010). Metadata (e.g., timestamp, tags, GPS) of the photos were used to generate past tourists' travel trails, which were then combined to generate travel itineraries for future tourists. De Choudhury *et al.* (2010) showed that high quality itineraries can be automatically constructed from Flickr data. However, again it is important to note that contextual information (except location) was not considered in these studies.

Summary. Research on trajectories has gained increasing interest in recent years, probably due to the increasing abundance of trajectory datasets in daily life. Most of the studies have focused on developing methods to understand moving objects' behavior based on their movement trajectories. Compared to these data mining applications, this research uses trajectories for generating recommendations in LBS.

In terms of using trajectories in LBS, different approaches have been proposed for creating novel applications, such as user, location, and activity recommendations. However, several limitations should be pointed out. Firstly, current approaches were often designed for and evaluated with a specific type of location history. These approaches might not be easily applied to other types of location histories. For example, the approach developed for GPS trajectories in Zheng *et al.* (2011) is hard to apply for trajectories constructed from social media. A uniform conceptual model for representing human behavior extracted from different kinds of location histories is still missing. Secondly, current approaches employing location histories for recommendations often fail to consider contextual information except the current location and user preferences. However, other contextual information, such as weather, time (weekday vs. weekend), and companion, might be potentially relevant for generating recommendations, especially location/place recommendations in LBS. It is still unclear how these additional contextual factors can be integrated into recommendation algorithms, and whether adding these additional contextual factors can help to improve recommendation quality.

In addition, existing approaches in the literature are often designed for and evaluated within a specific geographic scale, such as citywide or indoor (e.g., museum).

This dissertation attempts to address the above limitations. A methodology of deriving context-aware location recommendations from different kinds of location histories is proposed for LBS applications. We are interested in how contextual information can help to provide more relevant recommendations. The proposed methodology is evaluated with three real-word location history datasets, which reflect different scales of application scenarios and consist of different types of location histories: Delft city GPS dataset (city center), Vienna zoo GPS dataset, and trajectories constructed from Flickr photos uploaded for Vienna (citywide).

2.4 Related projects

This section provides a brief survey of LBS applications in a chronological order, with a focus on mobile guides. The systems were selected based on their technological innovation, scientific novelty, and relevance to our research. There were several reviews on mobile guides in the literature (Chen and Kotz 2000, Baus *et al.* 2005, Krüger *et al.* 2007, Raper *et al.* 2007b, Bohnert 2010, sec. 2.4). Here, we provide an up-to-date survey. More importantly, when describing these applications, we are especially interested in how they model users' interests and need, which additional contextual features they use, and how services are adapted to these kinds of information.

Hippie was an internet based museum guide which may be used in a stationary context at home and in a mobile scenario on the spot (Oppermann and Specht 1999). Content selection and presentation was adapted to the current devices, network connection, current location, as well as user knowledge and preferences (obtained from interaction history). A rule-based system and a domain hierarchy were exploited to determine appropriate content or recommend routes through the museum.

GUIDE was a handheld mobile guide for tourists (Cheverst *et al.* 2000). The context features used by GUIDE included the user's current location, profile, and the opening times of the attractions. Information about the city and attractions was tailored by utilizing the current context, e.g., the current location and the opening times of nearby attractions. GUIDE also employed a static user model obtained from explicit user input to generate a tailored city tour.

The ***CRUMPET*** project developed a personalized and location-based mobile guide (Schmidt-Belz *et al.* 2002). The context features used by CRUMPET were location and user interests. It employed an adaptive user model, which learnt user interests from the user's interaction with the system. Services were tailored to users' interests by using a domain taxonomy of tourist-related services.

Museum Wearable was a wearable computer for museum visitors, which provided a user-adapted audiovisual augmentation of the surrounding environment (Sparacino 2002). It built a progressively refined user model (with the help of a Bayesian network) from users' physical path in the museum and length of stops. The user model was then used to deliver a personalized audiovisual narration to the visitor.

COMPASS made location-based personalized POI recommendations to tourists (van Setten *et al.* 2004). Location was considered as a primary criterion to select relevant services in the near surroundings of the user. A user' interest model was manually initialized and further automatically updated by the system based on the user's feedback for specific POIs. A map view visualizing the current location and a selection of nearby POIs was presented to the user. An ontology describing the class hierarchy of POIs was employed for this purpose.

DTG (Dynamic Tourist Guide) provided personalized sightseeing tours in real-time by considering explicitly-stated static user interests, the available time period, and the current location (Hagen *et al.* 2005). An ontology was employed to automatically select relevant POIs.

ec(h)o was an audio museum guide acting as an augmentation of an existing exhibition installation (Wakkary and Evernden 2005). The soundscapes changed based on the position of the visitor in the space, the visitor's history with viewing the artifacts, and their individual interests in relation to the museum collection. Users' interests were first provided by themselves, and then updated by learning from their interaction and movement. Relevant content was retrieved with the help of an ontology and rule-based system.

CityVoyager recommended shops to mobile users by considering the current location and users' interests (Takeuchi and Sugimoto 2006). Users' interests (i.e., implicit ratings on shops) were estimated by analyzing their past location history acquired with GPS. An item-based collaborative filtering was employed to predict ratings for "nearby" (considering transition probability) shops. Those shops with higher ratings were recommended to the users.

GeoWhiz was a restaurant recommendation system for mobile users (Horozov *et al.* 2006). Restaurants were recommended based on the current location and users' interests. It proposed a location-enhanced collaborative filtering (CF) method: location was used as a key criterion to select nearby restaurants (e.g., within 1 km radius), and a traditional CF based on explicit ratings was used to rank these nearby restaurants.

UbiquiTO was an agent-based mobile guide for tourists (Cena *et al.* 2006). It adapted the content and its presentation according to the device used, users' preferences, and the context of information (i.e., the user location and time of the day). It kept tracking the user's behavior, and refined/updated the user model during the interaction. Some knowledge bases were employed for user modeling and service adaptation.

Park et al. (2007) proposed a personalized restaurant recommendation system. It provided personalized recommendations using a Bayesian network, which was defined by an expert and refined by using a training dataset. The network modeled the probabilistic influences of users' interests and context (e.g., season, time of a day, position, weather, and temperature) on the restaurant attribute values (class, price and mood).

The ***PEACH (Personal Experience with Active Cultural Heritage)*** project developed a multimedia mobile guide for museums (Stock and Zancanaro 2007). It also dealt with automatic adaptation of content presentation to different output devices and user interests. It combined both mobile and stationary systems in parallel, where the latter ones were used to show details. A user's interests were modeled based on both explicit feedback (e.g., by changing a different virtual character) and implicit observations of his/her interactions with the devices.

Li et al. (2009) proposed a multi-stage collaborative filtering method to provide event recommendations based on the current location and users' interests. Firstly, the Adaptive

Resonance Theory network was used to perform users' clustering, based on users' profiles which were explicitly provided by the users themselves. For each user, the sequential pattern mining method was used to extract individual sequential rules. The current location was used to predict a user's next places (by matching the sequential rules). Events for those next places were recommended to the users. In order to address the cold-start problem, cluster sequential rules were also extracted for each user cluster. Recommendations of new users were performed by matching their cluster sequential rules.

The ***Kubadji*** project aimed to provide personalized mobile guidance for museum visitors (Bohnert 2010). It recommended exhibits of interest, and provided personalized content delivery for these exhibits according to users' interests and current location. A user's interests were inferred from the non-intrusive observations of his/her behavior in the physical environment.

The ***GeoLife*** project developed several LBS applications for location and friend recommendations (Zheng, Xie, *et al.* 2010, Zheng *et al.* 2011). The recommendations were made by considering users' interests and location (current location and location history). Users' interests were modeled from their location histories, i.e., GPS trajectories. A user similarity measure was developed by using users' location histories. Users with high similarity to the current user were his/her potential friends. Locations visited by these potential friends were then recommended to the current user. For a detailed description, please refer to Section 2.3.2.

Bader et al. (2011) employed multi-criteria-decision-making methods to make gas station recommendations for car drivers. The recommendations were generated by considering the current time, location, gas level of the car, and prices. They used utility functions to model the importance of different context elements, which were derived from a preliminary user study.

Hung et al. (2012) combined a knowledge-based approach and collaborative filtering for making artwork recommendations in a museum. A visitor had to describe his/her interests before using the system. The initial user profile was then used in the knowledge-based approach for making recommendations. During the visit, users can also give ratings to the artworks, which were then employed to improve the recommendation.

Pombinho et al. (2012) recommended POIs by considering the users' previous interaction (i.e., searches) with the system, and therefore a content-based approach was employed. They allowed the user to define geographic areas (e.g., work area or home area) and used temporal distances to identify which POIs are open by the time the user gets there.

iConAwa provided a user with a list of nearby POIs and users who permit sharing of their location information according to his/her current location, time, and preferences (Yılmaz and Erdur 2012). Contextual information and POIs were modeled using ontology.

Yu and Chang (2013) developed a personalized mobile travel planning system. It employed a rule-based approach for making hotel, restaurant, and sightseeing spot recommendations.

Cheng *et al.* (2013) recommended places by using users' check-in history on social media. A user similarity measure was developed based on not only the positions of the check-ins but also their semantic categories, such as "shopping" and "eating". A user-based CF was then employed for making place recommendations.

Yang and Hwang (2013) made location recommendations by using ratings provided by other tourists. CF was employed for this purpose. Users were required to indicate their ratings on attractions. The system then employed mobile peer-to-peer communications for exchanging ratings between different users.

Table 2.1 summarizes and compares the main characteristics of the above systems. We mainly focus on the aspects of application domain, context features exploited, user profiling, and adaptation (prediction) mechanism employed, and adaptation level ("what is adapted?" e.g., content presentation and item recommendations).

Table 2.1 Comparison of related projects

	Domain	**Context**	**User profiling**	**Adaptation mechanism**	**What is adapted?**
Hippie	Museum guide	Location, user, device, network	Learning from interaction	Knowledge-based	Content selection, presentation

GUIDE	City guide	Location, user, time	Explicit user input	Knowledge-based	Recommend ation
CRUMPET	City guide	Location, user	Learning from interaction	Knowledge-based	Recommend ation
Museum Wearable	Museum	Location, user	Learning from interaction		Content presentation
COMPASS	City guide	Location, user	Explicit user input, Refining from interaction	Knowledge-based	Recommend ation, presentation
DTG	City guide	Location, user, time	Explicit user input	Knowledge-based	Recommend ation
ec(h)o	Museum guide	Location, user	Explicit user input, refining from interaction	Knowledge-based	Content presentation
CityVoyager	Shop recommenda tion	Location, user	Learning from interaction	Collaborativ e filtering	Recommend ation
GeoWhiz	Restaurant recommenda tion	Location, user	Learning from explicit ratings	Collaborativ e filtering	Recommend ation
UbiquiTO	City guide	Device, user, location, time	Refining from interaction	Knowledge-based	Content selection, presentation
Park et al. (2007)	Restaurant recommenda tion	Location, user, time, weather	Explicit user input	Knowledge-based	Recommend ation
PEACH	Museum guide	Device, user	Explicit user input, refining from interaction	Knowledge-based	Content presentation
Li et al. (2009)	Event recommenda tion	Location, user	Explicit user input	Collaborativ e filtering	Recommend ation
Kubadji	Museum guide	Location, user	Learning from interaction	Collaborativ e filtering	Recommend ation, content

					presentation
GeoLife	Location and user recommendation	Location, user	Learning from interaction	Collaborative filtering	Recommendation
Bader et al. (2011)	Gas station recommendations	Location, time, gas level of the car, prices		Knowledge-based	Recommendation
Hung et al. (2012)	Museum guide	Location, user	Explicit user input	Knowledge-based, collaborative filtering	Recommendation, content presentation
Pombinho et al. (2012)	Location recommendation	Location, time, user	Explicit user input, learning from interaction	Content-based	Recommendation
iConAwa	Location recommendation	Location, time, user		Knowledge-based	Recommendation
Yu and Chang (2013)	Location recommendation	Location, time, user	Explicit user input	Knowledge-based	Recommendation
Cheng *et al.* (2013)	Location recommendation	Location, user	Learning from users' check-ins	Collaborative filtering	Recommendation
Yang and Hwang (2013)	Location recommendation	Location, user	Explicit user input	Collaborative filtering	Recommendation

Some important insights can be observed from table 2.1.

1) Many mobile guides have provided users with location-aware and personalized services. However, many of them focused on adapting content presentation rather than location recommendations.
2) Some of the systems made recommendations by asking users to explicitly state their interests, e.g., GUIDE, DTG, GeoWhiz, Park *et al.* (2007), Li *et al.* (2009), Hung *et al.* (2012), Yu and Chang (2013), and Yang and Hwang (2013). It is important to note that, LBS users are often involved in

many tasks and activities during using mobile devices. Therefore, non-intrusive user modeling which does not require explicit user input should be introduced.

3) There were also mobile guides employing knowledge-based approaches for adaptive recommendations. These knowledge-based approaches often required a long underlying learning (knowledge acquisition) process and a good understanding of the application domain, both of which are very time-consuming and impractical for many LBS applications.
4) Systems employing CF for recommendations do not suffer from the problem of "knowledge acquisition bottleneck" (see Section 2.2.1). However, existing mobile guides of this kind only use location and user as context features, and fail to consider additional contextual information such as weather and companion.

In this dissertation, we investigate techniques for providing personalized and context-aware location recommendations in LBS that *(a) model users from their location histories (e.g., GPS trajectories), which can be considered as non-intrusive observations of their previous movements, (b) employ CF, and (c) exploit more contextual information (e.g., weather, companion, time) than location and user.* These are very promising: (1) LBS users prefer non-intrusive user modeling, as they are often involved in many other tasks and activities during their use of mobile devices. (2) With the increasing availability of GPS-enabled devices, more and more people start to record their travel/sports experience with GPS logs. In the meantime, with rapid advances in geotagged social media, recent years have also witnessed many people publishing their travel information and experiences via social media, such as Foursquare check-ins and Flickr photos. This "self-reported" information can be also used to construct users' location histories. These location histories may reflect people's travel experiences in the environment. Research has shown that experiences from past users (especially similar users) in similar contexts can help the current user efficiently solve their problems (Wexelblat 1999, Zheng *et al.* 2011). Therefore, these location histories provide an abundance of data for deriving recommendations. (3) Exploiting more contextual information might improve the recommendation quality. (4) CF-based methods do not require an explicit representation of domain knowledge, which is very welcome in real world LBS applications.

3 Implicit User Profiling from Location Histories

Collaborative filtering (CF) is probably the most popular and widely implemented recommendation technique (Adomavicius and Tuzhilin 2005). It recommends to a user the items that other users with similar tastes liked/used in the past (Resnick and Varian 1997). CF exploits information about existing users' behavior (e.g., usage) towards or opinions/feedback on items for predicting which items the current user will most probably like or be interested in. Items can be of any type, such as movies, books, products, or locations/places.

The first step of CF is to build user profiles from users' feedback/opinions on items made over time. In most of the existing CF applications, such as movie and product recommendations, a user profile is often represented as a set of ratings given by the user on different items, and each rating is modeled as a triple <user, item, rating>, e.g., <"Tom", "Titanic", 4>. Ratings can be explicit and implicit (Nichols 1997). Explicit ratings require explicit actions from users (e.g., indicating a rating on a scale) which bring some burden to them, and interrupt normal patterns of their action (Nichols 1997). Ratings can be also inferred from users' implicit feedback, such as moving tracks and interaction history. With the increasing availability of GPS-enabled devices, more and more people start to record their travel experiences as GPS trajectories. In the meantime, recent years have also witnessed many people publishing their travel information via social media, for example, Flickr photo uploading and Foursquare check-ins, which might be used to construct their location histories. When making location recommendations based on these location histories (recorded by GPS devices or constructed from the web), the user profile model described before (i.e., rating-based) is insufficient, as it cannot model users' behavior of visiting various locations, e.g., in which orders these locations are visited, and transit time between these locations.

Furthermore, in order to extract meaningful user profiles, methods on semantically processing raw location histories (e.g., GPS trajectories) should be developed. Due to their commonality in many trajectory applications, the concepts of stop and move, as proposed by Spaccapietra *et al.* (2008)), have been often employed. Different approaches have been proposed to extract stops and moves from raw trajectories (Alvares *et al.* 2007, Palma *et al.* 2008). However, as discussed in Section 2.3.1, these methods need to carefully calibrate the input parameters, whose optimized values are application-dependent, and sometimes should be personalized. For example, when visiting the Vienna zoo, the duration threshold that defines a stop in the panda house (Giant Panda) might vary for different tourists.

This chapter addresses the above two challenges, and aims to develop a methodology of implicit user profiling from location histories. Section 3.1 explores a model of contextual user profiles, which can be used to represent users' interests in various locations as well as behavior of visiting such locations. In order to extract meaningful user profiles from raw location histories, a duration-threshold-free SMoT (DTF-SMoT) and a stay-point-based SMoT (SP-SMoT) are developed in Section 3.2. As an improvement of the original SMoT method, DTF-SMoT only requires a set of pre-defined geographic areas as inputs. The duration thresholds, which are geographic area-dependent and user-dependent, are automatically learnt from the data. Similar to SMoT, DTF-SMoT is effective for scenarios with sufficient background geographic information. For scenarios where defining a complete set of candidate stops is rather difficult, we develop SP-SMoT, which improves the existing CB-SMoT by replacing its original input parameters with parameters that are more intuitive to understand and easier to configure and tune. We evaluate the proposed DTF -SMoT and SP-SMoT methods in Section 3.3, and summarize the results in Section 3.4.

DTF-SMoT and SP-SMoT can be used to semantically process location histories for extracting/building contextual user profiles. These methods can be combined with the context-aware CF methods developed in Chapters 4 and 5 to make personalized and context-aware location recommendations in LBS.

3.1 Contextual user profile model

In this section, we define some terms and propose a model of contextual user profile. This model can be used to represent a user's semantic location history, which can be extracted from her/his raw location histories, such as GPS trajectories.

3.1.1 Definitions

Definition 1 (Location history or trajectory). A raw location history (*LocH*) or trajectory is a sequence of points, each of which contains a location ($p_i.L$), and a timestamp ($p_i.T$). Thus, $LocH = p_1 \rightarrow p_2 \rightarrow \ldots \rightarrow p_n$, where $\forall 1 \leq i \leq n\text{-}1, p_i.T < p_{i+1}.T$. In this research, the terms "location history" and "trajectory" are used interchangeably.

Depending on the types of location history, $p_i.L$ might be very different. For example, in terms of GPS-based location histories (i.e., GPS trajectories), $p_i.L$ is a latitude/longitude pair. For Bluetooth trajectories, $p_i.L$ might be represented as a symbol (e.g., the ID of a Bluetooth scanner or beacon) or a relative location. For Foursquare check-ins, $p_i.L$ is a place or venue.

In order to have a meaningful interpretation of movement behavior, raw trajectories should be semantically processed. Spaccapietra *et al.* (2008) conceptualized a trajectory as a sequence of stops and moves. This conceptual view has been shown to be a useful framework for semantic trajectory processing in the literature (Palma *et al.* 2008, Bogorny *et al.* 2009, Andrienko *et al.* 2011, Yan *et al.* 2011, Renso *et al.* 2013, Rinzivillo *et al.* 2013). Similar to Spaccapietra *et al.* (2008), we define stops and moves as follows.

Definition 2 (Stop). A stop *s* is a sub-trajectory ($p_i \rightarrow p_{i+1} \rightarrow \ldots \rightarrow p_{i+k}$) of *LocH*, such that (i) the moving object does not move, as far as the application view of this trajectory is concerned; (ii) the temporal extent is a non-empty time interval; (iii) all stops in *LocH* are temporally disjoint. A stop $s = (Loc, arvT, levT, Dur)$, where $s.Loc = \{p_i, p_{i+1}, \ldots, p_{i+k}\}$, $s.arvT = p_i.T$, $s.levT = p_{i+k}.T$, and s.Dur = $p_{i+k}.T - p_i.T$.

Stops are application-dependent. A sub-trajectory is a stop in an application, but it might not be a stop in another application.

A stop happens at a place/location. It can be considered as a user's visit to this place/location. In this research, based on the information about other users' stops (visits) at various places/locations, we recommend places/locations for the current user to visit.

Definition 3 (Move). A move *m* is a sub-part ($p_i \rightarrow p_{i+1} \rightarrow \ldots \rightarrow p_{i+k}$) of *LocH*, such that the part is delimited by two extremities that represent either two consecutive stops, or p_1 and the first stop, or the last stop and p_n, or p_1 and p_n (if *LocH* has no stop). p_1 is the first point of *LocH*, while p_n is the last point of *LocH*. A move $m = (Loc, arvT, levT, Dur)$, where $m.Loc = \{p_i, p_{i+1}, \ldots, p_{i+k}\}$, $m.arvT = p_i.T$, $m.levT = p_{i+k}.T$, and $m.Dur = p_{i+k}.T - p_i.T$.

Definition 4 (Semantic Location history or user profile). An individual's semantic location history (*Sem_LocH*) is a sequence of stops and moves.

$Sem\_LocH = s_1 \xrightarrow{m_1} s_2 \xrightarrow{m_2} \ldots \xrightarrow{m_{n-1}} s_n$, where $\forall 1 \le i \le n-1,\ s_i.levT < m_i.arvT < m_i.levT < s_{i+1}.arvT.$

In many applications, stops and moves extracted from a trajectory can be enriched with semantics. For example, stops can be enriched with more information, such as where a stop is (e.g., name of its geographic area) and activity (i.e., what activity was carried out during stop). Moves are often labeled with speed and transportation mode. Different applications might have different attributes for stops and moves.

3.1.2 Context and contextual user profile model

Context is a multifaceted concept that has been studied in different disciplines, such as computer science, cognitive science, linguistics and psychology (Adomavicius and Tuzhilin 2011). In this research, we adopt the definition proposed by Dey (2001, p. 5): "Context is any information that can be used to characterize the situation of an entity. An entity is a person, place, or object that is considered relevant to the interaction between a user and an application". We also adopt a "representational" view of context as described in Dourish (2004). According to this view, contexts/situations that the system may be in can be characterized or differentiated by a set of observable attributes, such as "weather". These attributes can be regarded as context dimensions (parameters). Contextual dimensions and their possible values are known at the stage of system development, and the structure of them does not change over time. This view gives us a simple and intuitive definition for building accurate predictive models (Baltrunas 2011). In this research, each contextual dimension is represented as nominal types, e.g., the contextual dimension "*weather*" can have values of "sunny", "rainy" and "windy", dimension "companion" can be "alone", "with family", "with small kids", and "with others". Therefore, an attribute-value based structure is used to represent these contextual dimensions.

Definition 5 (Attribute and context parameter). An attribute *A* is a pair *(name, range)* where *A.name* is a unique label out of some name spaces, and *A.range* is the set of valid values that can be assigned to the attribute, also known as value range. Each contextual dimension (parameter) *CP* is an attribute.

Sometimes, values of a contextual dimension (e.g., temperature) are obtained from sensors, and represented as numeric data. As pointed out by many other authors, such as Coutaz *et al.* (2005) and Keßler *et al.* (2009), these numeric values are often mapped to nominal symbols for further applications. Therefore, without loss of generality, this research models context parameters as a nominal type, whose value range is an explicit enumeration of all allowed nominal values (e.g., an enumeration of "day_of_week {weekend, weekday}").

This research aims at providing context-aware location recommendations in LBS, i.e., recommending locations/places matching the context of a user's visit, which is defined as follows:

Definition 6 (Context model of visits). Context model of visits *CM* contains an ordered list of relevant context parameters, each of which is represented as an attribute. Thus, $CM = (CP_1, CP_2, \ldots, CP_n)$, where $n > 0$. The symbol $\widehat{CM}$ denotes the space of the context model. In other words, $\widehat{CM}$ is the set of all possible situations under which a visit can occur.

CM is application-dependent. A context parameter is considered as relevant in application A, but it might not be relevant in application B. For example, weather might be a relevant context parameter for outdoor travel activities, but it might not be relevant for indoor shopping. Therefore, one of the challenges of employing this "representational" view of context is the identification of relevant context parameters from a large set of potential context parameters. We will address this issue in Chapter 5.

Definition 7 (Context of a visit). The context of a visit $Context\_of\_visit \in \widehat{CM}$ is an instance of *CM*. $Context\_of\_visit = (v_1, v_2, \ldots, v_n)$, where $\forall 1 \le i \le n,\ v_i \in CP_i.range$.

Definition 8 (contextual user profile). A contextual user profile *C_LocH* is a pair of semantic location history, and its context of visit. Thus, *C_LocH = (Context_of_visit, Sem_LocH).*

Please note that, for simplicity, we also assume that the context of visit does not change during a user's visit.

To illustrate the model introduced above, an example of a contextual user profile is shown in Figure 3.1. A series of stops and moves can be extracted from each visit (sampled by raw trajectory), and each stop and move can be annotated with different attributes, such as duration of stops, transportation modes between stops, activities at stops. The whole visit happens within a specific context/situation, which is characterized by a set of relevant context parameters. For example, a situation can be <"weather: rainy", "companion: alone", "purpose: shopping">.

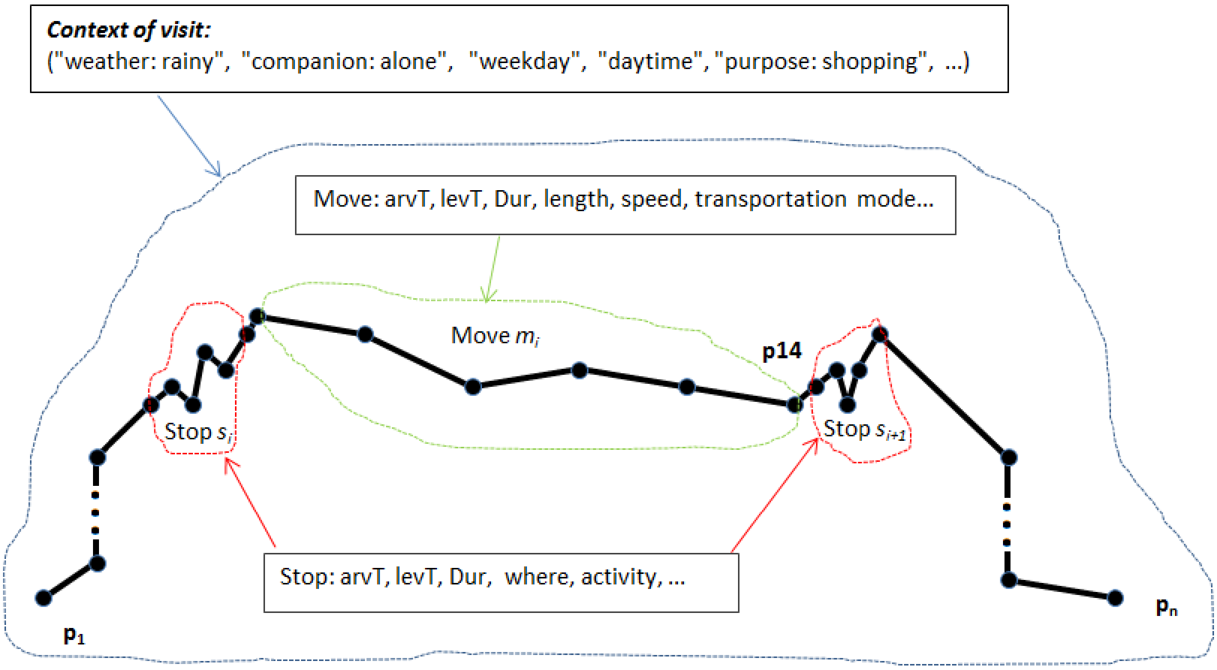


Figure 3.1 An example of contextual user profile: a user profile consists of a sequence of stops and moves. Its context_of_visit defines the situation under which the movement happened.

As discussed before, a contextual user profile consists of two parts: semantic location history, and its context of visit. The context of visit can be obtained from different sources. For example, "weather" can be obtained from the Internet, e.g., via Weather Underground API[15]; "Day of a week" and "time of a day" can be derived from raw trajectories or different sensors on mobile phones; "Companion" and "purpose" can be provided by users via questionnaires before asking for a recommendation. In

[15] http://www.wunderground.com/weather/api

order to extract semantic location history from raw trajectories, methods on semantically processing of raw trajectories should be developed. In the following, we mainly focus on GPS trajectories, which are the most common type of location histories.

3.2 Extracting semantic location history from raw trajectories

Raw location histories, e.g., GPS trajectories, often contain a sequence of positions. In order to have a meaningful interpretation, these data are often semantically annotated with background geographic information and other relevant information. In this research, we adopt the concepts of stop and move proposed by Spaccapietra *et al.* (2008) for semantic trajectory processing. We aim to extract users' semantic location history from their trajectories, which can be then used for making location/place recommendations in LBS.

As mentioned above, current state-of-the-art methods (e.g., SMoT and CB-SMoT) on extracting stops and moves from GPS trajectories require many input parameters, and these input parameters are hard to tune and optimize. In this section, two improvements of the existing methods are developed: a duration-threshold-free SMoT (DTF-SMoT) and a stay-point-based SMoT (SP-SMoT). DTF-SMoT (Section 3.2.1) is an improvement of SMoT method, and it frees users from defining a set of duration thresholds. On the other hand, SP-SMoT (Section 3.2.2) improves the existing CB-SMoT by replacing its original input parameters with parameters which are more intuitive to understand and easier to configure.

DTF-SMoT and SP-SMoT can be used to semantically process location histories for extracting/building contextual user profiles. These methods can be combined with the context-aware CF methods developed in Chapter 4 and Chapter 5 to make personalized and context-aware location recommendations in LBS.

3.2.1 Duration-threshold-free SMoT (DTF-SMoT)

Duration-threshold-free SMoT is an improvement of the SMoT method proposed by Alvares *et al.* (2007). In SMoT, a set of pre-defined geographic areas (i.e., "candidate stops", a polygon in the geographic space) and their minimum duration thresholds are required as additional inputs. If an object has stayed in an area for a duration longer than the time threshold, it is considered to have stopped in this area. The main challenge of this method is the definition of a suitable set of areas and their duration thresholds.

In order to reduce the number of parameters required, we develop a two-stage approach. This approach only requires a set of pre-defined geographic areas ("candidate stops") as inputs, and therefore, it is named as duration-threshold-free SMoT (DTF-SMoT). As mentioned above, each geographic area is represented as a topologically closed polygon ($\mathrm{a}_i \in R^2$, R denotes the set of real numbers, and R^2 denotes the two-dimensional geographic space) in the geographic space. Every polygon is disjoint (i.e., non-overlapping) with each other, i.e., $\forall\ 1 \leq i, j \leq n,\ i \neq j$, $\mathrm{a}_i \cap \mathrm{a}_j = \emptyset$, where n is the number of areas. The definition of this set of areas is often carried out domain experts by carefully studying the application domain, i.e., only the areas that are interesting to the application are included, e.g., only the areas being worth to visit are included. The definition of this set also needs to consider the GPS accuracy in the target environment, i.e., each geographic area should be defined appropriately to make sure that every user's visits to different areas can be differentiated with the current location-acquisition techniques. Therefore, a geographic area might contain one or more POIs, such as shops and restaurants. Recall that the aim of this research is to make personalized and context-aware location recommendations. These pre-defined geographic areas can be considered as candidate locations/places for recommendation.

In the following, we describe in detail the workflow of the DTF-SMoT algorithm. At the first stage, DTF-SMoT continually verifies the intersection of the trajectory with this set of polygons, and extracts a set of intersections. At the second stage, for each intersection, we determine whether it is a stop or not by comparing the current duration (cur_Dur) with a duration threshold (Δ_{dur}). Δ_{dur} is geographic area-dependent and user-dependent. It is dynamically determined by the characteristics of the user itself (reflected by $userAvgDur$ – the average duration of the user at all the pre-defined areas she/he visited) and the characteristics of the intersected area (reflected by the ratio of $areaAvgDur$ and $AvgDur$). $areaAvgDur$ stands for the average duration of other users at this area, and $AvgDur$ is the average duration of all users at all the areas they visited.

We observe that $\frac{cur\_Dur}{userAvgDur}$ can be considered as the user's *normalized* duration at the current geographic area/polygon (i.e., normalized by the average duration of the user at all the areas she/he visited), and $\frac{areaAvgDur}{AvgDur}$ is the *normalized* average duration at this geographic area (i.e., normalized by the average duration of all users at all the areas they visited). The difference between $\frac{cur\_Dur}{userAvgDur}$ and $\frac{areaAvgDur}{AvgDur}$ measures whether the current user spent (relative to her/his average duration) more and less time at the geographic area than an average user. If the user spent more time at the area than an average user (i.e., $\frac{cur\_Dur}{userAvgDur} > \frac{areaAvgDur}{AvgDur}$, or $cur\_Dur > userAvgDur * \frac{areaAvgDur}{AvgDur}$), she/he might be more likely to have stopped in this area. Therefore,

$$\Delta_{dur} = userAvgDur * \frac{areaAvgDur}{AvgDur} \qquad \text{(Eq. 3.1)}$$

Algorithm 3.1 summarizes the DTF-SMoT algorithm. We first tag all the points in the trajectory as move point "M". Then, for each point, we check whether it is in one of the pre-defined geographic areas or not. If yes, the point is tagged as the name of the pre-defined area. At the second stage (lines 12-21), we check each sub-trajectory with a same tag other than "M". If the duration in this sub-trajectory is bigger than the dynamic duration threshold (Δ_{dur}) obtained from the get_duration_threshold function, this sub-trajectory is a stop, otherwise, all the points in this sub-trajectory will be tagged as "M". During the process, all the intersections and corresponding duration will be stored in a database on the server to allow effective computation of $areaAvgDur$ and $AvgDur$ (as required in the get_duration_threshold function) for new trajectories (see line 11).

After these two stages, each user's semantic location history can be extracted from her/his trajectory. Currently, the algorithm performs two scans in the second stage. However, it is possible to combine them together for better performance.

As an improvement of SMoT, DTF-SMoT automatically learns a set of duration thresholds, which are geographic *area-dependent* and *user-dependent*, from the data. Similar to SMoT, DTF-SMoT is suitable for scenarios where defining a complete list of candidate stops is possible. In the following, we explore approaches to discover stops that are unknown *a priori*, but may be potentially interesting to the application.

Algorithm 3.1: Duration-threshold-free SMoT (DTF-SMoT)

```
Input: a trajectory Traj ={p1, p2, ..., pn}, a set of geographic areas A {a1, a2, ..., an}
Output: a semantic location history Sem_LocH = {SL1, SL2, ..., SLn}, where SLi is
            either a stop or a move
i=1;
tag all the points in Traj as move point "M";
WHILE (i<= size (Traj)) DO
    IF (∃ aj∈A && p_in_polygon (pi.L, aj)) THEN
        enterPoint = i; i++;
        WHILE (p_in_polygon (pi.L, aj)  DO
            i++;
        i--;  leavePoint =i;
        tag all the points between enterPoint and leavePoint as the name of aj;
    i++;
Store all the intersections and corresponding durations into database
FORALL the consecutive points with the same tag other than "M"DO
    get the duration cur_dur of these points;
    area_name = tag of these points;
    Δdur = get_duration_threshold (Traj, area_name);
    IF (cur_dur >= Δdur) THEN
        stop = (pfrom, pto, enterTime, area_name, duration); Sem_LocH.add (stop);
    ELSE
        change all these points'tags to "M"
FORALL the consecutive points with the same tag "M" DO
    move = (pfrom, pto, enterTime, duration); Sem_LocH.add (move);
Re-order all the elements in Sem_LocH according to their enterTime.
Return Sem_LocH;
```

3.2.2 Stay-Point-based SMoT (SP-SMoT)

The stay-point-based SMoT (SP-SMoT) method is based on the concept of stay point proposed by Zheng *et al.* (2011). A stay point (region) stands for a geographic region where a user stayed over a certain time interval. Therefore, the extraction of a stay region depends on a distance threshold θ_{dist} and a duration threshold θ_{dur}. In SP-SMoT, firstly, we extract stay regions from a trajectory with an algorithm similar to the one in Zheng *et al.* (2011). Then, SP-SMoT verifies for each stay region if it intersects the pre-defined candidate stops (polygons). If a stay region does not intersect any of the pre-defined polygons, it is still regarded as an interesting place, and tagged as "unknown stop". Otherwise, the intersection between the stay region and each intersected candidate stop is extracted and expanded to include neighboring points that are also in the candidate stop. If the duration of the expended sub-trajectory is longer than the pre-defined duration threshold of the candidate stop, a stop annotated with the name of the candidate stop is created. All the other parts of the trajectory will be considered as moves. Algorithm 3.2 depicts the SP-SMoT algorithm.

Algorithm 3.2: Stay-Point-based SMoT (SP-SMoT)

Input: a trajectory Traj $=\{p_1, p_2, \ldots, p_n\}$, a distance threshold Δ_d, a time threshold Δ_t, a set of geographic areas A $\{a_1, a_2, \ldots, a_n\}$, and a set of time thresholds for these area T $\{t_1, t_2, \ldots, t_n\}$

Output: a semantic location history Sem_LocH = $\{SL_1, SL_2, \ldots, SL_n\}$, where SL_i is either a stop or a move

```
i=1;
tag all the points in Traj as move point "M";
WHILE (i<= size (Traj)) DO
    j =i+1;
    WHILE j<size(Traj)  DO
        IF (distance (p_i, p_j)> Δ_d) THEN
            IF (time_difference(p_i, p_j)>Δ_t) THEN
                tag all the points between p_I and p_j as stay point "SP";
            i=j; BREAK;
        j++;
FORALL the consecutive points with the same tag "SP" DO
    FORALL  a_k where a_k ∈ A && Length(Intersection(a_k, cur_subtrajectory))>0 DO
        Intersect_traj= Intersection(a_k, cur_subtrajectory);
        extend Intersect_traj with neighboring points that are also in a_k;
        get the duration Dur of the cur_subtrajectory;
        IF(Dur >=t_k) THEN  //if the duration is longer than the threshold
            stop = (p_from, p_to, enterTime, a_k.name, Dur); Sem_LocH.add (stop);
            tag all the points between p_from and p_to as "S";
        ELSE
            tag all the points between p_from and p_to as "M";
    IF cur_subtrajectory does not intersect with any candidate stop THEN
        uk_stop = (p_from, p_to, enterTime, "unknown", Dur); Sem_LocH.add (uk_stop);
        tag all the points between p_from and p_to as "S";
FORALL the consecutive points with the same tag "M" DO
    move = (p_from, p_to, enterTime, duration); Sem_LocH.add (move);
Re-order all the elements in Sem_LocH according to their enterTime.
Return Sem_LocH;
```

The proposed approach is very similar to CB-SMoT (Clustering-based SMoT) proposed by (Palma *et al.* (2008). However, they differ from each other with regard to two aspects. Firstly, CB-SMoT uses density-based clustering to identify low-speed clusters of a trajectory, while SP-SMoT employs a distance threshold θ_{dist} and a duration threshold θ_{dur} to extract stay regions. We believe that SP-SMoT would be more interesting, as the required parameters (i.e., duration/distance threshold) are more intuitive to understand and easier to configure. Secondly, CB-SMoT maps each low-speed cluster to either an unknown stop or a known stop, while each stay region in SP-SMoT can consist of several known stops or an unknown stop. In real world applications, a low-speed cluster might contain several stops. In these senses, SP-SMoT would be more appropriate than CB-SMoT.

Compared to DTF-SMoT, SP-SMoT can discover stops that are unknown *a prior*. If two or more unknown stops intersect with each other, they can be merged together, and receive the same name. Therefore, SP-SMoT is more useful for scenarios where defining a complete list of candidate stops is difficult (e.g., the lack of sufficient domain information).

In many applications including mobile guides, knowing users' presence (or absence) at different places/locations is very important. From this sense, both DTF-SMoT and SP-SMoT are very useful as they can identify the places/locations a user has visited. In this research, based on the information about other users' stops (visits) at various places, we recommend places for the current user to visit.

3.3 Evaluation and discussions

In this section, we discuss some experimental evaluations to study the performance of the proposed DTF-SMoT and SP-SMoT methods on extracting stops and moves from GPS trajectories. The datasets used for the experiments are discussed in Section 3.3.1. In Section 3.3.2, we describe the experimental setting, and discuss the results.

3.3.1 Datasets

Two real world GPS datasets were used for experimental evaluations, and they reflected different scales of application scenarios: 1) contextual GPS dataset collected from Vienna zoo (Tiergarten Schönbrunn, Vienna, Austria); 2) contextual GPS dataset in Delft city center (Netherlands).

1) Vienna zoo dataset

In cooperation with Vienna zoo (Tiergarten Schönbrunn), we collected trajectories in the zoo in 2011. At the gate of the zoo, we approached visitors, and invited them to carry GPS loggers with them while walking through the zoo. Different GPS loggers were used in the data collection, such as QStarz BT-Q1000XT[16] and Blumax GPS-4044 Datalogger[17]. Visitors were told to put the logger in their bag or

[16] http://www.qstarz.com/Products/GPS%20Products/BT-Q1000XT-F.htm

pocket. Before they start, we recorded some additional information about their visits, such as having an annual pass ("Y" or "N"), with whom ("alone", "with children", or "others"), weather ("rainy" or "sunny/cloudy"). A small gift (e.g., a pen) was given to them after returning the GPS loggers.

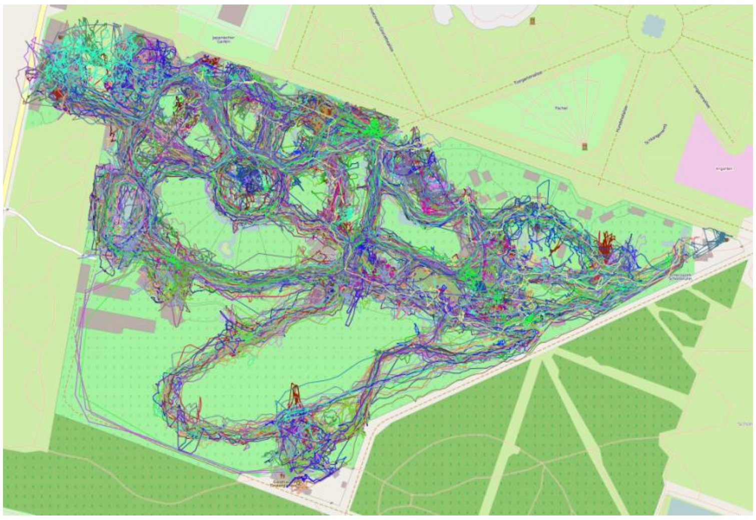


Figure 3.2 Visualization of the cleaned GPS dataset collected from Vienna zoo (Map data: OpenStreetMap and Contributors, CC-BY-SA)

In total, we collected 209 valid trajectories of all kinds of visitors in different situations. As mentioned above, raw GPS trajectories are often very noisy. A simple mean smoothing was applied to clean these trajectories. We also removed GPS points that were outside of Vienna zoo, as users cannot be outside of the zoo during the visit. Figure 3.2 visualizes these cleaned trajectories using the open source GIS software Quantum GIS[18]. This visualization uses OpenStreetMap[19] as the background base map. A vague outline of the road network in the zoo can be seen from this visualization.

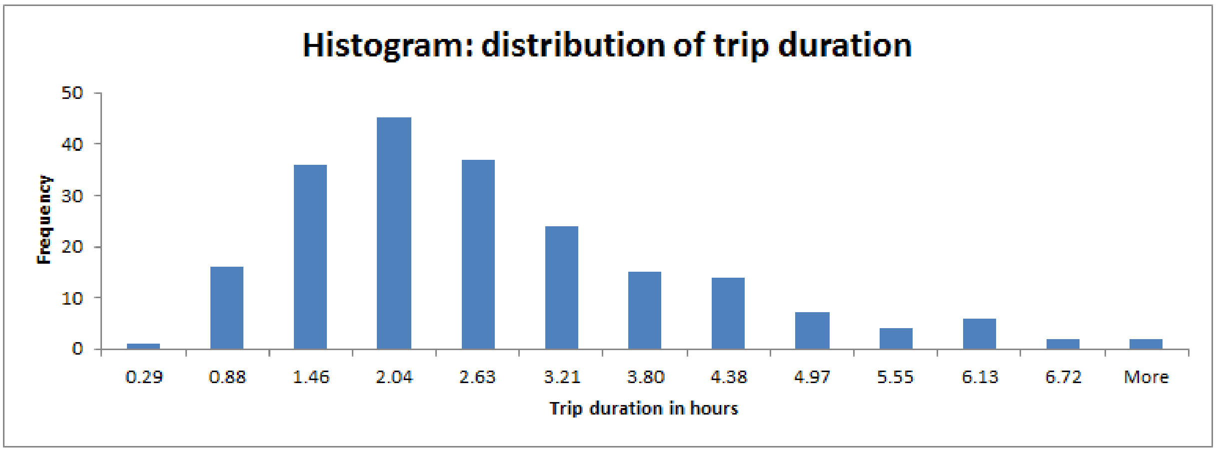
In order to have a closer look at the dataset, we checked the duration and length profile of these trajectories. On average, the trajectories have a length of 3,393 meters, and duration of 9,596 seconds (~2.67 hours). Figure 3.3 shows the distribution of duration and travel distance of these

[17] http://www.blu-max.com/products/gps_4044_logger.html

[18] http://www.qgis.org/

[19] http://www.openstreetmap.org/

trajectories.



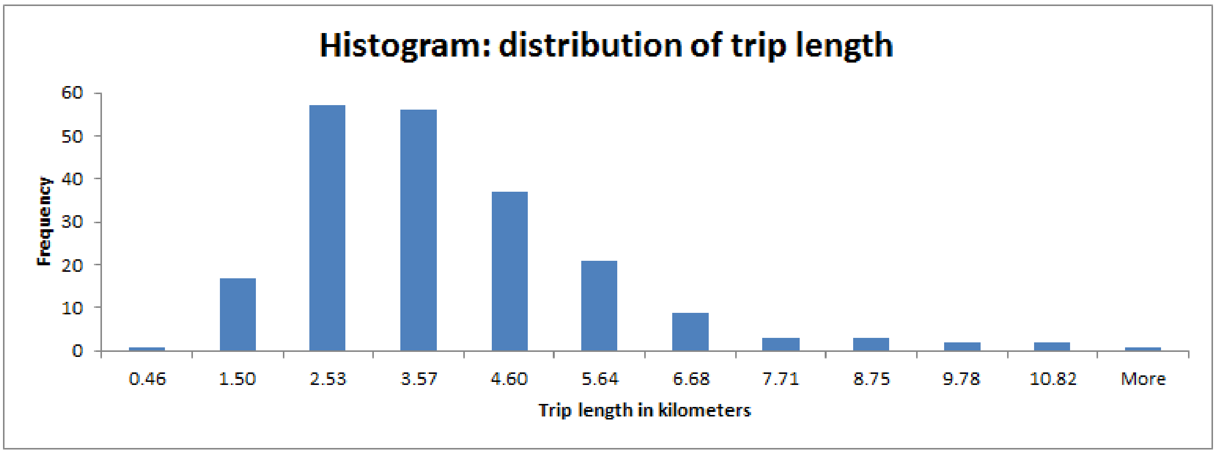


Figure 3.3 Distributions of trip duration and trip length (Vienna zoo dataset)

2) *Delft city dataset*

This dataset was shared by Prof. dr. ir. S.C. van der Spek from Delft University of Technology, who tracked visitors in the city center of Delft (the Netherlands). Prof. van der Spek and his team collected this dataset from 18 November to 21 November 2009. They handed out GPS loggers (e.g., Qstarz BT-Q1000X Travel Recorder) to participants at one of the two parking facilities, located on the south side (i.e., Zuipoort parking) and west side (i.e., Phoenix parking) of the Delft city center. To understand the behavior better, a questionnaire was filled in by participants on returning the GPS loggers. The questionnaire included demographic data (age, home, family status, and profession), the purpose of the trip (e.g., shopping, tourism, leisure, and other) and the frequency of visiting Delft city center. Additionally, the weather condition was recorded by the data collection team.

A total number of 325 participants (= the number of GPS tracks) participated in the research. We validated this dataset, and removed trips with inconsistent GPS track and questionnaire. Finally, 255 trips were valid. In order to clean up the trajectories, a simple mean smoothing was applied. A visualization of all these cleaned trajectories using Quantum GIS is presented in Figure 3.4, with OpenStreetMap being the background base map.

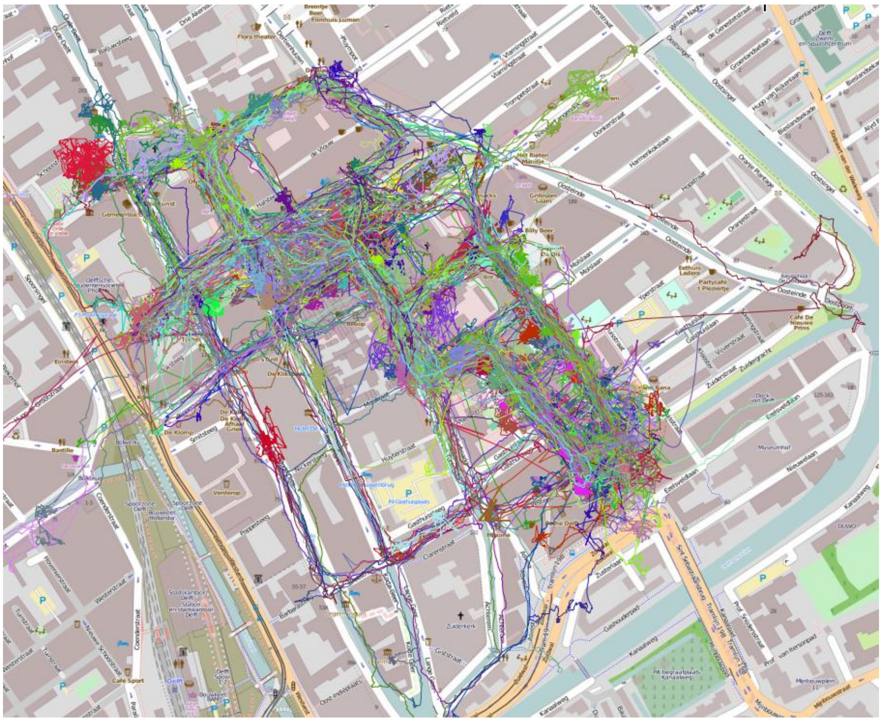
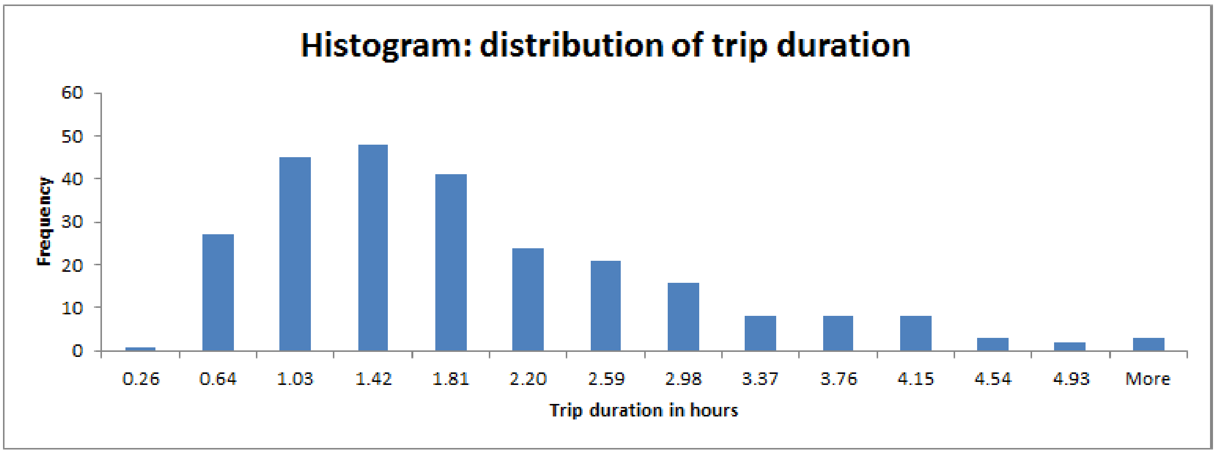


Figure 3.4 Visualization of the cleaned GPS trajectories collected from Delft city center (Map data: OpenStreetMap and Contributors, CC-BY-SA)

In order to have a closer look at the dataset, we checked the duration and length of these trajectories. On average, the trajectories have a length of 2,776 meters, and duration of 6,299 seconds (~1.75 hours). Figure 3.5 shows the distribution of duration and travel distance (length) of these trajectories.



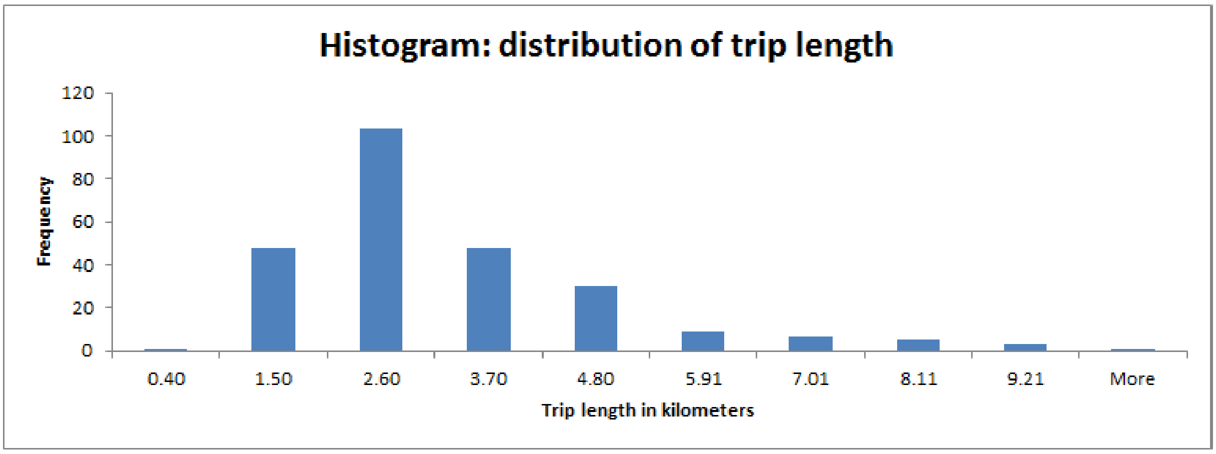


Figure 3.5 Distributions of trip duration and trip length (Delft city dataset)

3.3.2 Experimental setting, results and discussion

We used the real-world datasets described above to evaluate the performance of the proposed DTF-SMoT and SP-SMoT methods in extracting stops and moves from GPS trajectories. Specifically, we compared these two methods with a state-of-the-art method, i.e., the original SMoT method proposed by Alvares *et al.* (2007). SMoT is a benchmark method for extracting stops and moves if a set of candidate stops and their duration thresholds can be defined. We were interested in the following two questions:

1) Does DTF-SMoT, which requires fewer parameters as inputs, achieve similar results to SMoT?
2) Can SP-SMoT discover known stops as SMoT while also discovering unknown stops?

In order to measure how much the results obtained from these methods are similar, we designed a sequential similarity measure based on the Longest Common Subsequence (LCS). LCS is the longest subsequence (not necessarily consecutive) common to all sequences in a set of sequences (often just two) (Bergroth *et al.* 2000). A longer LCS implies a more similar value between two sequences. Therefore, the LCS-based similarity between sequences extracted by SMoT and DTF-SMoT/SP-SMoT was measured as:

$$Sim(Seq_{SMoT}, Seq) = \frac{2 \times |LCS|}{|Seq_{SMoT}| + |Seq|} \qquad \text{(Eq. 3.2)}$$

Where $|LCS|$ is the number of stops in the LCS, $|Seq_{SMoT}|$ and $|Seq|$ are the number of stops in the sequence extracted from SMoT and that from DTF-SMoT/SP-SMoT. The similarity value of 1 means the results are identical, while 0 means the results do not have any overlap. Please note that the LCS measure considers not only the stops (i.e., stop places and duration) but also the sequential relationships of these stops[20].

All these methods require a set of pre-defined geographic areas ("candidate stops", i.e., polygons) as inputs. We defined this set of areas by carefully studying the layouts and GPS accuracy of both scenarios (Vienna Zoo and Delft city center)[21]. Therefore, each area might contain one or more POIs[22], such as shops and restaurants. The final sets of areas used for the Vienna zoo dataset and the Delft city dataset are depicted in Figures 3.6 and 3.7 respectively.

[20] This measure can be improved by considering both false positives and true negatives as introduced in the field of information retrieval (Salton and McGill 1986).

[21] For example, we visited all the POIs defined in the official Vienna zoo map (http://www.zoovienna.at), and aggregated some of the close POIs as a single area according to the GPS accuracy.

[22] As mentioned above, this is due to the constraints of current location-acquisition technologies, which make it not always possible to identify the exact POI a user is visiting.

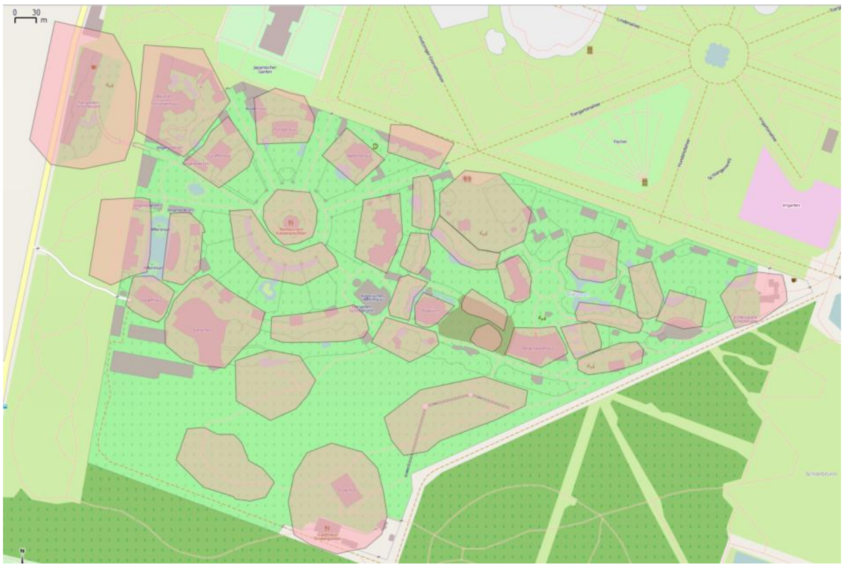


Figure 3.6 Candidate stops for the Vienna zoo dataset

Figure 3.7 Candidate stops for the Delft city dataset

In addition to the set of areas, both original SMoT and SP-SMoT require a set of duration thresholds for all the pre-defined areas. We empirically defined these duration thresholds according to the size of the pre-defined areas as well as the types of areas[23]. For SP-SMoT, two additional parameters (i.e.,

[23] For example, we asked three university students (two females and one male, all are new to the zoo) to visit all the pre-defined areas in Vienna zoo and recorded the duration they visited each area. We then averaged these students' duration to approximate the duration threshold for each area.

a distance threshold θ_{dist} and a duration threshold θ_{dur}) are also required as inputs. θ_{dist} and θ_{dur} play a key role in detecting stops, and their optimal values are application dependent. Derived from the layout of the zoo and average walking speeds of tourists in the zoo, θ_{dist} and θ_{dur} were empirically set to 30 meters and 90 seconds for Vienna zoo dataset. In other words, if a user stays over 90 seconds within a distance of 30 meters (i.e., walking speed lower than 0.33 m/s, which is comparable to the average walking speeds of tourists in the zoo 0.35 m/s), a stay point is detected. Similarly, we set θ_{dist}=25 and θ_{dur}=60 for the Delft city dataset. In the future, we will investigate the sensitivity of these parameters in detail and identity means to calibrate these parameters in an automatic manner.

1) Results from the Vienna zoo dataset

Table 3.1 shows the comparisons of DTF-SMoT, SP-SMoT and the original SMoT for the Vienna zoo dataset. We compared these methods with regard to the aspects of input parameters, the number of (known) stops detected, the number of unknown stops detected, and similarity of the results.

Table 3.1 Comparisons of DTF -SMoT, SP-SMoT and SMoT (Vienna zoo dataset)

Algorithm	**Inputs (in addition to trajectories)**	**# of (known) stops detected**	**# of unknown stops detected**	**Similarity with SMoT results**
SMoT	A set of candidate stops (polygons), their duration thresholds	2648	-	-
DTF-SMoT	A set of candidate stops (polygons)	2762	-	0.93
SP-SMoT	A set of candidate stops (polygons), their duration thresholds, a distance threshold θ_{dist}, a duration threshold θ_{dur}	2470	79	0.92

Comparisons of DTF-SMoT and SMoT: As can be seen from Table 3.1, DTF-SMoT has found more stops than the original SMoT. This is because the duration threshold for each candidate stop in DTF-SMoT is dynamically computed, and the duration of stop detected by DTF-SMoT is no longer enough to be considered as a stop by SMoT. However, on average, the results obtained from DTF-SMoT are about 93% similar with the results computed by SMoT. In other words, even with fewer parameters as inputs, DTF-SMoT still achieves comparable results as SMoT.

Comparisons of SP-SMoT and SMoT: SP-SMoT has detected fewer known stops than the original SMoT. This is because stops in SP-SMoT are extended from stay points, and the duration of some stops detected by SMoT is no longer enough to be considered as stay points. Table 3.1 also shows that on average, the results obtained from SP-SMoT are about 92% similar with the results computed by SMoT. We also observe that the number of unknown stops detected by SP-SMoT is relatively small (i.e., ~0.38 unknown stops per trajectory), this is due to the fact that the candidate stops, which were carefully designed, were dense and covered most of the parts of Vienna zoo.

2) Results from the Delft city dataset

Similarly, for the Delft city dataset, we compared DTF-SMoT, SP-SMoT and the original SMoT with regard to input parameters, the number of (known) stops detected, the number of unknown stops detected, and similarity of the results. Table 3.2 shows the comparisons.

Table 3.2 Comparisons of THF-SMoT, SP-SMoT and SMoT (Delft city dataset)

Algorithm	Inputs (in addition to trajectories)	# of (known) stops detected	# of unknown stops detected	Similarity with SMoT results
SMoT	A set of candidate stops (polygons), their duration thresholds	2283	-	-
DTF-SMoT	A set of candidate stops (polygons)	2172	-	0.90
SP-SMoT	A set of candidate stops (polygons), their duration thresholds, a distance threshold θ_{dist}, a duration threshold θ_{dur}	2033	203	0.87

Table 3.2 shows that the results obtained from DTF-SMoT are about 90% similar with the results

computed by SMoT. This confirms the results of the Vienna zoo experiment that even with fewer parameters as inputs, DTF-SMoT still achieves comparable results to SMoT. We also observe that the number of unknown stops detected by SP-SMoT is relatively small (i.e., ~0.80 unknown stops per trajectory), this is also because the candidate stops were dense and covered most of the parts of the study area.

In summary, the evaluation results of both datasets are similar. The results demonstrate that even with fewer parameters as inputs, DTF-SMoT still achieves comparable results to SMoT. On the other hand, SP-SMoT can discover known stops as SMoT while discovering unknown stops.

The current evaluation can be improved by using experiments with human participants. For each participant, we can compare how well DTF-SMoT, SP-SMoT and SMoT can identify stops reported by him/her. As DTF-SMoT uses personalized duration thresholds to identify stops, we expect that DTF-SMoT will generate much better results than the state-of-the-art method, i.e., the original SMoT. We will investigate this issue in the future.

3.4 Summary and conclusions

Building user profiles from users' feedback/opinions on items made over time is the first step when making CF recommendations. This chapter explored a methodology of implicit user profiling from location histories. Specifically, we proposed a model of contextual user profiles to represent users' interests in various locations (reflected by their visits to these locations and the duration of these visits) as well as behavior of visiting such locations, which can be implicitly derived from users' location histories, such as GPS trajectories and trajectories constructed from Foursquare check-ins. A duration-threshold-free SMoT (DTF-SMoT) and a stay-point-based SMoT (SP-SMoT) were then developed to semantically process location histories for extracting/building these contextual user profiles.

We evaluated the proposed methods with two real-world GPS datasets: Vienna zoo dataset and Delft city dataset. Our evaluation shows that for both datasets, DTF-SMoT, which requires fewer parameters as inputs, achieves comparable results to the state-of-the-art method, i.e., the original SMoT method proposed by Alvares *et al.* (2007). Therefore, DTF-SMoT is a good replacement of

SMoT for scenarios where duration thresholds for candidate stops are rather difficult to define. On the other hand, SP-SMoT achieves comparable results to SMoT approach, however, it can also discover stops that are unknown *a priori*, but may be potentially interesting to the application. Therefore, SP-SMoT is more useful for scenarios with insufficient background geographic information, in which defining a complete list of candidate stops is difficult.

In the following two chapters, we develop context-aware collaborative filtering (CF) methods based on the model of contextual user profiles proposed in this chapter. These CF methods can be integrated with the DTF-SMoT and SP-SMoT methods to derive personalized and context-aware location recommendations from location histories for LBS applications.

4 Personalized Location Recommendations from Location Histories

Recent research has recognized that as LBS users are often involved in many tasks and activities during using mobile devices, non-intrusive recommendations, which are derived from the non-intrusive observations of users and do not require users' explicit inputs, should be introduced into LBS (Filippini Fantoni 2003, Bohnert 2010). This chapter discusses a collaborative filtering (CF) method for deriving personalized location recommendations from a large number of users' location histories, which can be considered as non-intrusive observations of these users' previous movements. We are interested in how other users' interests in various locations (reflected by their visits to these locations and the duration of these visits) and motion behavior of visiting such locations, as extracted from their location histories (as in Chapter 3), can be utilized to provide the current user with relevant location recommendations. Please note that, the CF method proposed in this chapter does not use contextual information like weather and companion (with whom), and therefore, it can be considered as a non-contextual CF method. This method will be enriched with contextual information in Chapter 5 to provide users with personalized and context-aware location recommendations in LBS.

As discussed in Chapter 2, there was research focusing on deriving personalized recommendations from location histories or interaction histories (Han and Cho 2006, Takeuchi and Sugimoto 2006, Bohnert 2010, Zheng *et al.* 2011). Our research differs from these methods mainly on two aspects. Firstly, our methods are developed based on the concepts of stops and moves (see Section 2.1), which provide a fundamental and common framework for semantically processing different kinds of location histories, such as GPS trajectories and trajectories constructed from Flickr photos. In this sense, our method is not restricted to a specific kind of location history. Secondly, we investigate whether considering the order in which places/locations are visited, location popularity, duration at locations and transit time between locations contributes to the improvement of recommendation quality.

This chapter is structured as follows. Section 4.1 provides some notations and defines the problem. Section 4.2 develops a novel user similarity measure by considering the sequence property of movement (i.e., the order in which locations/places are visited), location popularity, duration at locations and transit time between locations. This measure is then used in Section 4.3 to identify users who are similar to the current user, and "opinions" (i.e., movements) of these similar users are further aggregated to generate personalized location recommendations for the current user. We evaluate our approaches in Section 4.4 and Section 4.5, and summarize this chapter in Section 4.6.

4.1 Notations and problem definition

In order to define the recommendation task, we need to introduce some notations. The set of users in the system will be denoted by U, and the set of items (i.e., locations/places in this research) by I. According to definition 4 in Chapter 3, for any user $u \in U$, his/her visit to these items can be modeled as a sequence of stops and moves, $Sem\_LocH^u = P_1^u(ST_{P_1^u}^u) \xrightarrow{M_{P_1^u, P_2^u}^u} P_2^u(ST_{P_2^u}^u) \xrightarrow{M_{P_2^u, P_3^u}^u} \dots \xrightarrow{M_{P_{m-1}^u, P_m^u}^u} P_m^u(ST_{P_m^u}^u)$, which can be considered as his/her user profile. $\forall\ 1 \le i \le m, P_i^u \in I$. $ST_{P_1^u}^u$ denotes the amount of time that u spends at P_1^u, and $M_{P_1^u, P_2^u}^u$ denotes the duration of the movement between P_1^u and P_2^u. We also use I^u to denote the subset of places/locations that have been visited by the user u. Please note that user profiles can be extracted from raw location histories by employing the DTF-SMoT and SP-SMoT methods in Chapter 3.

In this chapter, we mainly focus on personalized location recommendations, e.g., recommend a location to visit next. The problem consists in finding, for a particular user u and his/her current location (i.e., P_m^u), the new location $p \in I - I^u$ that u is most likely to visit.

CF, which recommends to a user the items that other users with similar tastes liked/used in the past, is probably the most popular recommendation technique (Adomavicius and Tuzhilin 2005). Among different CF methods, neighborhood-based CF (user-based and item-based) has gained a large popularity because of its simplicity, justifiability (easy to explain the reason behind prediction), efficiency (less computation and memory cost, suitable for mobile application scenarios), and abilities to provide serendipitous recommendations (Desrosiers and Karypis 2011). As a result, in this research, user-based CF is employed for deriving personalized location recommendations from location histories. Two important steps exist in user-based CF: identifying users who are similar to the current user with the help of user similarity measures (Section 4.2), and aggregating "opinions" of these similar users for making recommendations (Section 4.3).

4.2 User similarity measure

The key in CF is to locate other users whose "opinions" (i.e., movements in this research) can be used for generating recommendations for the current user. In this research, we identify these users in terms of their similarities with the current user, and similar users are defined as users with similar interests in various locations and similar motion behavior of visiting such locations.

As discussed in Chapter 3 and Section 4.1, a user profile consists of a sequence of stops and moves, and can be represented as:

$$Sem\_LocH^u = P_1^u(ST_{P_1^u}^u) \xrightarrow{M_{P_1^u,P_2^u}^u} P_2^u(ST_{P_2^u}^u) \xrightarrow{M_{P_2^u,P_3^u}^u} \dots \xrightarrow{M_{P_{m-1}^u,P_m^u}^u} P_m^u(ST_{P_m^u}^u).$$

In the following, we explore a user similarity measure for comparing the profiles of any two users. In line with Bohnert (2010) and Zheng *et al.* (2011), a user's visit[24] to a place/location (represented as a stop) and duration at this location can be used to approximate his/her implicit interest rating for the location; while his/her motion behavior of visiting different locations is reflected by the aspects of sequence relationships of locations visited (e.g., the order in which locations were visited), and transit time between locations. Therefore, a user similarity measure based on users' interests in various locations/places and motion behavior is developed, by considering sequence relationships of locations visited, location popularity, duration at locations, and transit time between locations.

[24] Please recall that, a user's visit to a place is identified if he/she stays at this place longer than a duration threshold. The literature has shown that the amount of time that a user spends at a place often correlates positively with preference and interest in this place (Froehlich *et al.* 2006, Bohnert 2010).

1) Considering sequential relationships of locations visited

Suppose there are three places A, B and C, users *a* and *b* visit them in the order of A -> B -> C, and user *c* visits them in the order of B -> A -> C. It is obvious that user *a* is more similar to *b* than to *c*. In this sense, we consider the sequential relationships of locations visited when measuring user similarity. In fact, the sequence property of users' movement is one of the vital characteristics differentiating location recommendations in LBS from other recommendations, such as movie recommendations or product recommendations (Zheng *et al.* 2011).

In the literature, different methods have been proposed for measuring trajectory similarity in terms of sequential relationship, such as the Longest Common Subsequence (LCS) approach in Yan and Zeng (2009), and Edit Distance on Real Sequence approach in Chen *et al.* (2005). In this research, the LCS approach is used, as it enables us to consider other aspects, such as location popularity, and duration at locations. It finds the longest subsequence (not necessarily consecutive) common to all sequences in a set of sequences (Bergroth *et al.* 2000). For example, the LCSs between G->C->A->T and G->A->C are GA and GC.

A longer LCS implies a higher user similarity value when only considering sequence relationship. Therefore, the user similarity considering sequence relationship is measured as:

$$Seq\_USim(a,b) = \frac{2\times|LCS|}{|Sem\_LocH^a|+|Sem\_LocH^b|} \times \frac{2\times|LCS|}{(Gap^a+1)+(Gap^b+1)} \quad \text{(Eq. 4.1)}$$

$|LCS|$ is the number of locations/places in the LCS, $|Sem\_LocH^a|$ and $|Sem\_LocH^b|$ are the number of locations/places visited by users a and b. Gap* is the index difference between the LCS's first location and the LCS's last location in each trajectory. The second part of the above measure gives a higher value when the LCS is consecutive in the trajectories.

Please note that, several LCSs might exist for two sequences. Therefore, we use the LCS achieving the highest similarity value.

2) Considering location popularity

It is obvious that two users accessing a set of locations visited by a few people might be more correlated than others who share a set of locations accessed by many people (Zheng *et al.* 2011). For instance, many people have visited the Stephansdom and Schönbrunn Palace, two well-known landmarks in Vienna (Austria). It might not be the case that all these people are similar to each other. However, if two users visited a location/place, which is not very popular, they might indeed share some similar preferences.

As a result, location popularity is considered when measuring similarity between two users. Different measures like entropy and inverse document frequency (IDF) have been proposed in the field of information theory. Due to its simplicity, IDF is employed in this research to model the popularity of a location. IDF is often used in information retrieval and text mining to measure whether a term (e.g., word) is common or rare across all documents. It is obtained by dividing the total number of documents by the number of documents containing the term, and then taking the logarithm of that quotient (Salton and McGill 1986). Similarly, we measure the popularity of a location *p* as:

$$IDF_p = \log\frac{N}{N_p} \qquad \text{(Eq. 4.2)}$$

where N is the number of all users, N_p is the number of users who visits location *p*.

IDF of each location is then used as a weighting factor and added to Eq. 4.1. Therefore, the similarity considering both sequence relationship and location popularity is measured as:

$$Seq\_Idf\_USim(a,b) = \frac{2\times\sum_{p\epsilon LCS} IDF_p}{\sum_{p\epsilon I^a} IDF_p + \sum_{p\epsilon I^b} IDF_p} \times \frac{2\times|LCS|}{(Gap^a+1)+(Gap^b+1)} \qquad \text{(Eq. 4.3)}$$

3) *Considering duration at locations (viewing time)*

The amount of time that a user spends at locations/places often correlates positively with preference and interest in locations (Bohnert 2010). Users spending similar time at places might be more similar to each other than those who do not. However, viewing time is also positively correlated with the complexity of the item being visited (Dean 1996, Johnston 1998). Based on the above consideration, we measure the similarity between two users' duration at a same location *p* as follows:

$$Dur\_Sim(p,a,b) = \begin{cases} 1, \left(\frac{ST_p^a}{ST_p} \geq 1 \; and \; \frac{ST_p^b}{ST_p} \geq 1\right) or \left(\frac{ST_p^a}{ST_p} \leq 1 \; and \; \frac{ST_p^b}{ST_p} \leq 1\right) \\ 0, \; else \end{cases} \qquad \text{(Eq. 4.4)}$$

where ST_p is the average duration at location *p* considering all users, and it reflects the complexity of the location.

$\frac{ST_p^a}{ST_p}$ can be regarded as user a's normalized duration at the current location. $\frac{ST_p^a}{ST_p} > 1$ means that user a spends more time at the location *p* than an average user, and therefore, user a might be more likely to like this location. If both users spend more (or less) time at *p* than an average user, they might all like (or dislike) the current location. Therefore, they are similar when considering duration at this location. Please note that Eq 4.4 simply maps the similarity value to either 1 or 0. It can be further improved by mapping the similarity value to a continuous scale between 1 and 0. In the evaluation, we show that even with this simple mapping (Eq 4.4), the results are already promising.

Eq 4.4 is then extended to measure the duration similarity for locations commonly visited by both users.

$$Dur\_USim(a,b) = \frac{\sum_{p \epsilon LCS} Dur\_Sim(p,a,b)}{|LCS|} \qquad \text{(Eq. 4.5)}$$

Finally, we combine Eq 4.3 and Eq 4.4 to measure user similarity, considering sequential relationships, location popularity, and duration at locations. We consider Dur_USim as a weight for LCS_IDF_USim, and therefore, multiplication instead of addition is applied in Eq 4.6[25].

$$Seq\_Idf\_Dur\_USim(a,b) = LCS\_IDF\_USim(a,b) \times Dur\_USim(a,b) \qquad \text{(Eq. 4.6)}$$

4) Considering transit time between locations

Users having similar transit between locations are more similar to each other than those who do not. For example, suppose that users a and b travel from A to B by foot, and user c travels from A to B by car. It is obvious that user *a* is more similar to *b* than to *c*. In this sense, we consider transit between locations when measuring user similarity. Here, we mainly focus on transit time between locations, i.e., movement duration between locations.

Similarly, we measure the similarity between two users' transit time between the same pair of

[25] Another reason is that we try to avoid finding suitable weights, which is required when using addition.

locations $p \rightarrow q$ as follows:

$$Tran\_Sim(p,q,a,b) = \begin{cases} 1, & \left(\frac{T_{p,q}^{a}}{T_{p,q}} \geq 1 \; and \; \frac{T_{p,q}^{b}}{T_{p,q}} \geq 1\right) or \left(\frac{T_{p,q}^{a}}{T_{p,q}} \leq 1 \; and \; \frac{T_{p,q}^{b}}{T_{p,q}} \leq 1\right) \\ 0, & else \end{cases} \quad \text{(Eq. 4.7)}$$

where $T_{p,q}$ is the average transit time from location *p* to location *q*.

Again, if both users spend more (or less) time to travel from *p* to *q* than an average user, they are similar when considering transit time between these locations.

Eq 4.7 is then extended to measure the transit similarity of two users.

$$Tran\_USim(a,b) = \frac{\sum_{p \rightarrow q \epsilon LCS} Transit\_Sim(p,q,a,b)}{|LCS|-1} \quad \text{(Eq. 4.8)}$$

Finally, we combine Eq 4.6 and Eq 4.8 to measure user similarity, considering sequential relationships, location popularity, duration at locations, and transit time between locations.

$$Seq\_Idf\_Dur\_Tran\_USim(a,b) = LCS\_IDF\_Dur\_USim(a,b) \times Tran\_USim(a,b) \quad \text{(Eq. 4.9)}$$

With Eq. 4.9, we can measure the similarity between any two users by comparing their user profiles, as extracted from their location histories.

4.3 Location recommendations

In this section, we use the user similarity measure (Eq. 4.9) to identify users who are similar to the current user (i.e., the one asking for recommendations), and aggregate these similar users' "opinions" (movements) for making personalized location recommendations, e.g., recommend a location/place to visit next.

Assume that the current user *u* has visited a set of locations/places. Currently he/she is at the location *p*, and asking "which place to visit next". The steps of the recommendation are as follows:

1) Identifying users whose next location after visiting *p* has not been visited by the current user;
2) Identifying the N users who are most similar to the current user *u*, using the user similarity measure in Eq. 4.9;

3) For the N most similar users, aggregating their next locations after visiting *p* (weighted by the user similarity values);
4) Selecting the location with the highest predicted value, and recommending it to *u*.

The above algorithm consists of one parameter N – the number of similar users (i.e., neighborhood size). We investigate the sensitivity of this parameter in Section 4.5.

In the following sections, we evaluate the method for personalized location recommendations with three real-world location history datasets. Section 4.4 describes the datasets for the evaluation. Section 4.5 presents and discusses the results of the experiments.

4.4 Datasets

In order to have a comprehensive evaluation of the proposed methods, the two real-world GPS trajectory datasets (Delft city dataset and Vienna zoo dataset) introduced in Chapter 3, as well as an additional trajectory dataset constructed from Flickr photo stream for the city of Vienna (for short, Flickr dataset) were used for the experimental evaluations. These three datasets reflected different scales of application scenarios and consisted of different types of location histories. In the following, we briefly describe how we processed these datasets.

4.4.1 Delft city dataset

As mentioned in Section 3.3, this dataset consisted of GPS trajectories and some additional information such as users' demographic data and purpose of the trip. For more details about the dataset, please refer to Section 3.3. In this chapter, we mainly aggregate the movement tracks for making personalized recommendations, while contextual information will be considered in Chapter 5.

After data cleaning, 255 GPS trajectories were valid. We then used the duration-threshold-free SMoT (DTF-SMoT) as proposed in Section 3.2.1 to extract a sequence of stops and moves from each GPS trajectory. As required in the DTF-SMoT method, a set of candidate stops and their boundaries (polygons) were defined (see Figure 3.7) by carefully studying the layouts of Delft city center and GPS accuracy. As mentioned in Section 3.1, the sequence of stops and moves extracted from a trajectory can be considered as the profile of the user who generated the trajectory. We evaluated the above CF methods for personalized location recommendations using these profiles.

4.4.2 Vienna zoo dataset

As mentioned in Section 3.3, the Vienna zoo dataset also consisted of GPS trajectories as well as some contextual information about users and their visits such as demographic data and weather. After data cleaning, 209 GPS trajectories were valid. Similar to what we did for the Delft city dataset, we then used the DTF-SMoT method to extract a sequence of stops and moves from each GPS trajectory. As required in the DTF-SMoT method, a set of candidate stops and their boundaries were defined (see Figure 3.6) by carefully studying the layouts of the zoo and GPS accuracy.

4.4.3 Flickr dataset

Currently, with the rapid advances in geotagged social media, location histories about people's travel can be also constructed from their "self-reported" information on the Internet, such as Foursquare check-ins and Flickr photos. The Flickr dataset was of this kind, and was constructed from the photos uploaded for the city of Vienna during January 2007 and January 2011 on the Flickr website.

Flickr APIs were used to retrieve photos within the boundary of Vienna. For each photo, different metadata were extracted, such as the owner of the photo, date uploaded, date taken, and latitude/longitude. In total, metadata of 154,343 geotagged photos were extracted. We cleaned the dataset according to some heuristic rules proposed in De Choudhury *et al.* (2010), and removed photos with inaccurate timestamps. We also filtered out photos from Viennese residents. We employed the heuristic rule proposed in De Choudhury *et al.* (2010) to differentiate tourists and residents. The rule was based on the assumption that while most tourists concentrate their visits within a short time period for several days, residents tend to take pictures of the city over a much longer period of time. Therefore, the differentiation of tourists and residents can be done by checking the span of the taken times between a user's first and last photos. Similar to De Choudhury *et al.* (2010), we set the time span threshold as 21 days.

In order to map photos to the locations visited, we used the list of Top 30 popular locations (landmarks, tourist attractions) compiled and published by the Vienna Tourist Board (Wiener Tourismusverband) according to the ticket sales statistics of 2009[26]. This list also contains the latitude/longitude information for each location. Similar to De Choudhury *et al.* (2010), we associated a photo to a location *p* whenever *p* is the closest location to the photo, and it was taken within 100 meters of *p*. The mapping results were then integrated with timestamps to construct a travel trajectory for each user, and further refined as a sequence of stops and moves according to the model of user profile in Section 3.1. We also filtered out user profiles with less than six locations visited. In total, we got 112 user profiles.

In the dataset, sometimes it is impossible to compute the duration at a location, if there is only one photo mapped to the location. We therefore did not extract duration at locations for each profile.

4.5 Evaluation and discussions

This section reports on the evaluation of the proposed method for personalized location recommendation, against some benchmarking methods. In Section 4.5.1, we describe the experimental setting. Section 4.5.2 discusses the sensitivity analysis of relevant parameters. In Section 4.5.3, we present and discuss the evaluation results of the three real-world datasets. Section 4.5.4 summarizes and discusses the results.

4.5.1 Experimental setup

Objectives. For the experimental evaluation, we aimed to investigate the effectiveness of the proposed properties (i.e., the order in which locations are visited, location popularity, duration at locations and transit time between locations) in the improvement of recommendation quality. In addition, we performed a sensitivity analysis on the number of similar users (neighborhood size) N to

[26] http://en.wikipedia.org/wiki/Tourist_attractions_in_Vienna

study its impact on the recommendation quality.

Benchmarking methods. In order to address the above objectives, we designed the following benchmarking methods for location recommendations:

1) Distance-based approach (in short, “Dist”): it recommends the closest location/place that has not been visited by the current user. This method was designed based on the observation that when no/little knowledge about a place is available, LBS often recommend to users with places that are close to the current location.
2) Set-based approach (in short, “Set”): instead of using the similarity measure $Seq\_Idf\_Dur\_Tran\_USim$ (Eq. 4.9), it measures the similarity between by comparing the locations visited by them, without considering the sequential relationships between locations. Specifically, the user similarity in this set-based approach is defined as:

$$Set\_USim\ (a, b) = \frac{2*\left|Sem\_LocH^{a} \cap Sem\_LocH^{b}\right|}{\left|Sem\_LocH^{a}\right| + \left|Sem\_LocH^{b}\right|} \qquad \text{(Eq. 4.11)}$$

   Where $\left|Sem\_LocH^{a} \cap Sem\_LocH^{b}\right|$ denotes the number of locations/places commonly visited by users a and b.
3) Sequence-based approach (in short, “Seq”): instead of using the $Seq\_Idf\_Dur\_Tran\_USim$, it measures user similarity by only considering sequence relationships between locations.
4) Seq + Idf: It measures user similarity by only considering sequence relationships between locations, and location popularity (computed as IDF).
5) Seq + Dur: It measures user similarity by only considering sequence relationships between locations, and duration at locations.
6) Seq + Tran: It measures user similarity by only considering sequence relationships between locations, and transit time between locations.
7) Seq + Idf + Dur: It measures user similarity by only considering sequence relationships between locations, location popularity, and duration at locations.
8) Seq + Idf + Tran: It measures user similarity by only considering sequence relationships between locations, location popularity, and transit time between locations.
9) Seq + Dur + Tran: It measures user similarity by only considering sequence relationships between locations, duration at locations, and transit time between locations.

Datasets. The three datasets described in Section 4.4 were used for the experiments. We only considered user profiles with at least six locations/places. In total, for the Delft city dataset, we had 116 user profiles (the number of locations visited per profile: M=8.8, SD= 2.4, range 6-18). For the Vienna zoo dataset, we had 167 user profiles (the number of locations visited per profile: M=11.9, SD= 4.5, range 6-27). For the Flickr dataset, we had 112 user profiles (the number of locations visited per profile: M=7.4, SD= 1.4, range 6-12).

Evaluation metric. Precision and recall are the most popular metrics for evaluating information retrieval systems (Salton and McGill 1986). Herlocker *et al.* (2004) pointed out that recall is an impractical measure in a recommendation system, due to the difficulty of identifying all the items that are relevant to the recommendation task. Therefore, we employed precision to evaluate the recommendation methods. In all the methods, we only recommended the top one location to the current user. Therefore, precision is either 1 or 0, depending on whether the recommended location is actually visited immediately by the current user or not[27]. We averaged the precision values for each method to represent its recommendation quality. In other words, the recommendation performance of each method was measured as the ratio of the number of successful recommendations (i.e., the recommended location is actually visited immediately by the current user) and the number of recommendation processes. In short, it can be considered as the percentage of successful recommendations.

Evaluation framework. We used a leave-one-out validation: For example, we trained all the recommendation methods on 115 of the 116 users for the Delft city dataset, and tested them on the remaining user. For each remaining user, we made recommendations starting from their fourth location, i.e., we did not make recommendations for the first four locations of each visit.

All the recommendation methods (except “Dist”) have one parameter to calibrate, i.e., the number of similar users (neighborhood size) N. In the experiments, we first implemented a sensitivity analysis to study the impact of neighborhood size N on recommendation quality. We then evaluated the

[27] The literature has shown that the amount of time that a user spends at a place often correlates positively with preference and interest in this place (Froehlich *et al.* 2006, Bohnert 2010). In other words, users tend to pass through a place quickly if they do not like this place. In our evaluation, a user’s visit to a place is identified if he/she stays at this place longer than a duration threshold. With this, we neglect the places users passing through quickly (e.g., places they do not like). In other words, we only extract places that users visited, and probably liked or were happy with. Therefore, if the recommended location is the same as the user’s next visited location, we consider the recommendation process as successful.

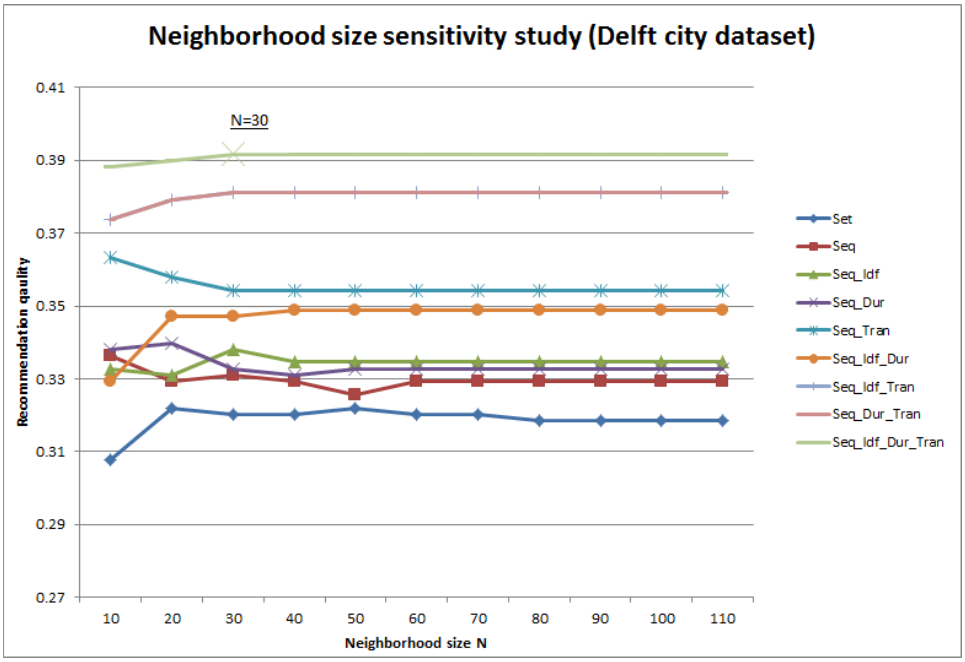
performance of the proposed method by comparing it with the benchmarking methods. We only used the parameter values achieving the best results for the comparison.

Therefore, the experiments can help us to answer the following questions:

1) What is the impact of the neighbor size N on the recommendation quality? Can the optimized parameter value be learnt from the training datasets?
2) Does considering other people's "opinions", sequential relationship, location popularity, duration at locations, and transit time between locations contribute to the improvement of recommendation quality?

4.5.2 Sensitivity analysis

All the recommendation methods (except "Dist") have one parameter to calibrate, i.e., the number of similar users (neighborhood size) N. Research has shown that the neighborhood size has a significant impact on the recommendation quality (Herlocker *et al.* 1999, Sarwar *et al.* 2000). In order to determine the effect of neighborhood size, we performed some experiments by varying the sizes. The results are shown in Figure 4.1. Due to the missing duration information in the Flickr dataset, we only compared the *Set*, *Seq*, *Seq_Idf*, *Seq_Tran*, *Seq_Idf_Tran* methods for this dataset.



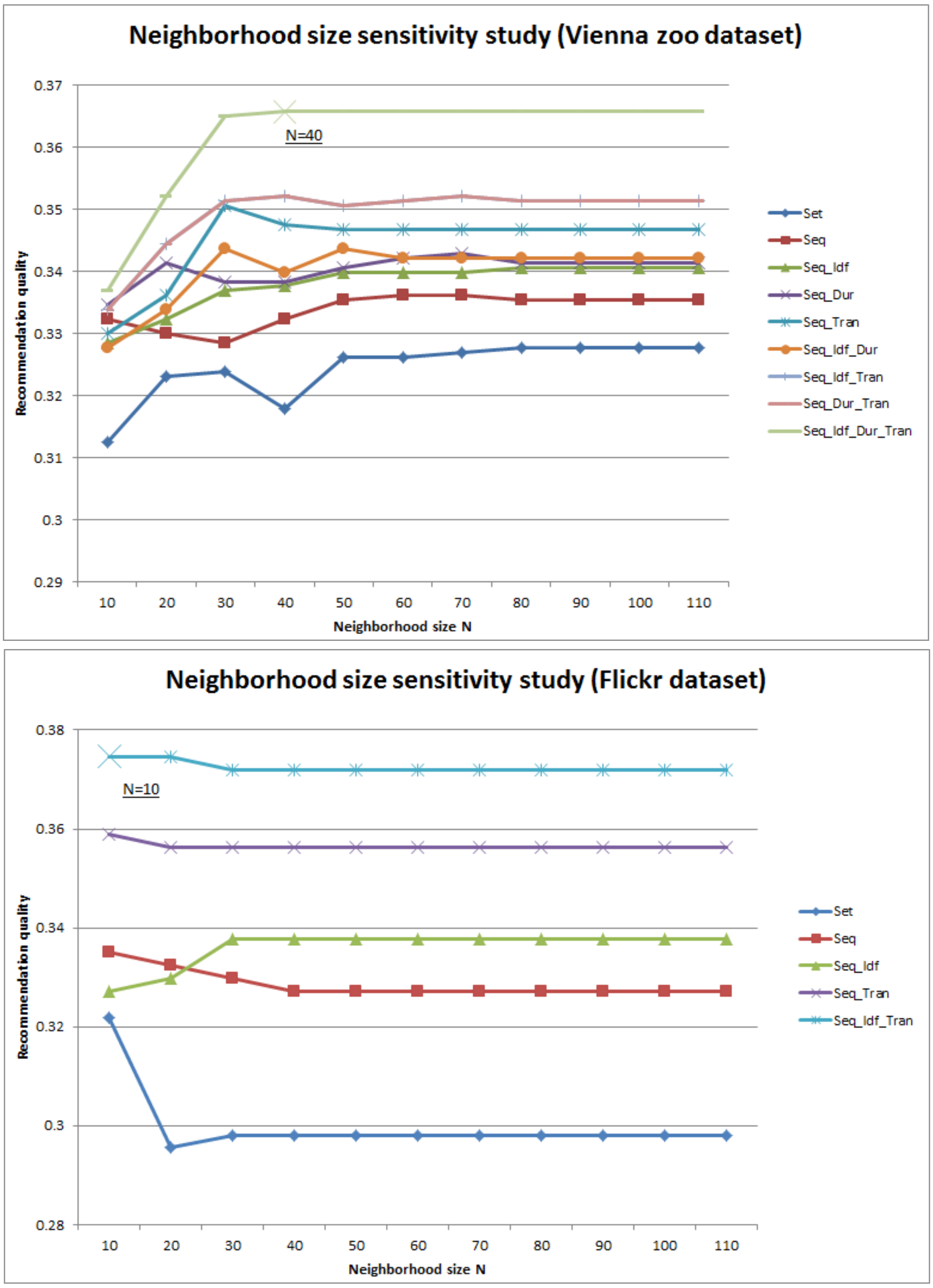


Figure 4.1 Impact of neighborhood size on recommendation quality

Figure 4.1 shows that the neighborhood size does affect the recommendation quality of all the methods. An interesting observation is that after a certain point, the recommendation quality becomes stable for all the methods. This is probably due to the way we aggregated similar users' movements. Recall that at the last two steps of the methods (see Section 4.3), we aggregated every similar user's next location/place after visiting the current location, by considering its similarity value with the current user. For each user, only a small group of users have higher similarity values with him/her, and all the other users have a similarity value closer to 0. Therefore, after a certain point, the recommended location remains the same, and the recommendation quality becomes stable.

Another interesting observation is on the *Set* method in the Flickr dataset, whose performance decreases when the neighborhood size changes from 10 to 20. This might be due to the increasing "options" (i.e., candidate locations) when having more neighbors. More work should be done on this aspect. This result also suggests that finding a suitable neighborhood size is very important for CF methods.

We also observe that the optimal number of neighbors is dataset dependent. In the Delft city dataset, the proposed Seq_Idf_Dur_Tran reaches its peak after 30, in the Vienna zoo dataset the peak is reached after 40, whereas in Flickr dataset the proposed Seq_Idf_Tran reaches its peak at 10. Given this fact, it is important to see if we can accurately estimate the optimal number of neighbors using the training dataset alone. One way of doing this is to perform a sensitivity analysis of the neighborhood size on the training datasets only (Sarwar *et al.* 2000). We focused on the best performing methods only (i.e., the Seq_Idf_Dur_Tran method for the Delft city dataset and the Vienna zoo dataset, and the Seq_Idf_Tran method for the Flickr dataset). Figure 4.2 shows the results.

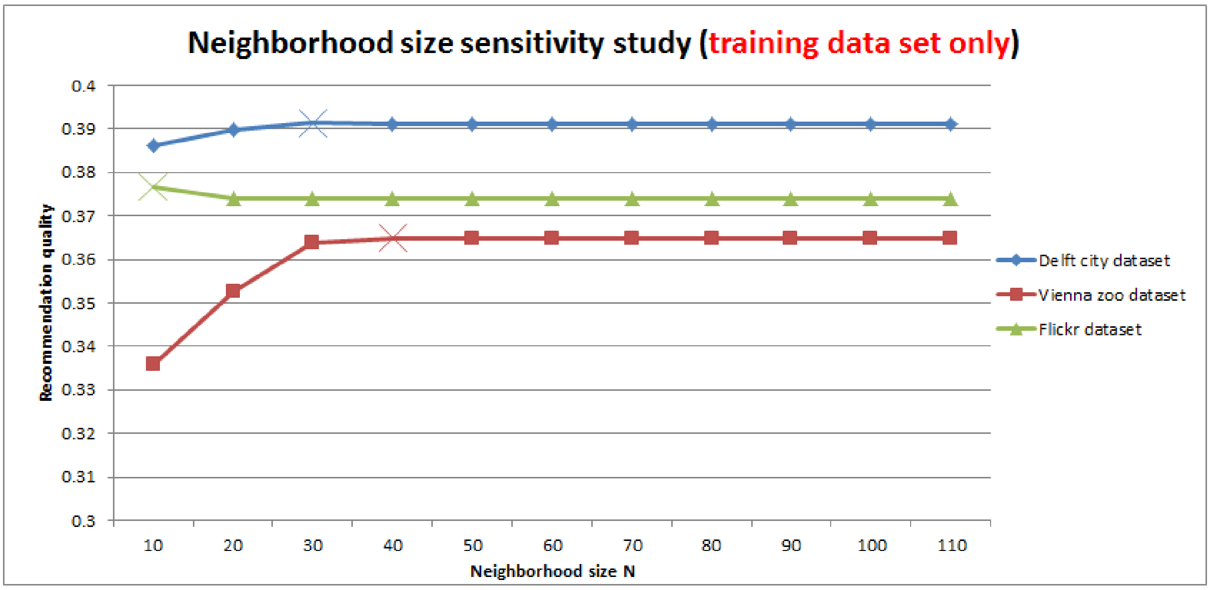


Figure 4.2 Impact of neighborhood size on recommendation quality of the best performing methods. The experiment was done by only using the training datasets.

Comparing Figures 4.1 and 4.2, the impact of neighborhood size on the recommendation quality of the best performing methods is similar for both cases. More importantly, the first peaks in Figure 4.2 are the same as those in Figure 4.1. Therefore, *the optimal number of neighbors can be correctly learnt from the training dataset alone*.

For the rest of the experiments in this section, we used a neighborhood size of 30 for the Delft city dataset, that of 40 for the Vienna zoo dataset, and that of 10 for the Flickr dataset.

4.5.3 Evaluation results

In order to investigate the effectiveness of the proposed properties (i.e., the order in which locations/places are visited, location popularity, duration at locations and transit time between locations) on improving the recommendation quality, we designed an experiment comparing the Seq_Idf_Dur_Tran method and all the benchmarking methods. All statistical tests in the following were one-tailed paired t-tests at the significance level $\alpha = 0.05$.

1) Evaluation with the Delft city dataset

The Delft city dataset was collected in Delft city center, which is a typical urban scenario. Figure 4.3 depicts the comparison of all the methods, which used different similarity measures.

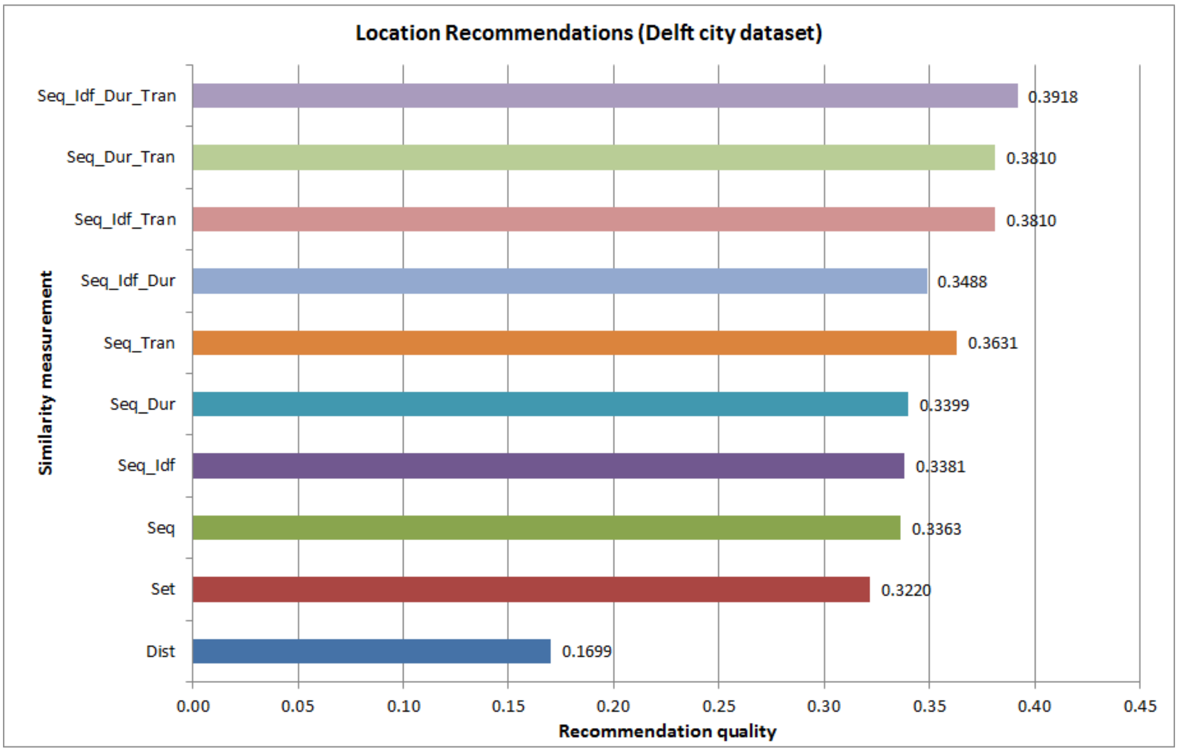


Figure 4.3 Comparison of the recommendation quality among different similarity measures (Delft city dataset)

Figure 4.3 shows that the proposed user similarity measure which considers the aspects of sequence relationships of locations visited, location popularity, duration at locations, and transit time between locations achieves the best result. In the following, we investigate each aspect/property in detail.

Considering other people's movements: Among all the methods, *Dist* recommends nearby place/location, while Set and Seq make location recommendations by considering the movements of other like-minded users. Figure 4.3 shows that both *Set* and *Seq* achieve significant better recommendation results than *Dist* ($p<0.001$), with improvements of 0.15 and 0.17 respectively. Therefore, considering other people's movements can help to improve recommendation quality. The results are consistent with the finding of Bohnert (2010) and suggest that other users' movements are better predictors of a user's movements than the distances between locations.

Considering the orders in which locations are visited (sequential relationships): *Seq* can be considered as an improvement of *Set* by considering sequential relationships. Figure 4.3 shows that *Seq* achieves better results than *Set*. However, the difference is only close to significant ($p=0.08$).

Considering location popularity: *Seq_Idf*, *Seq_Idf_Dur*, *Seq_Idf_Tran* and *Seq_Idf_Dur_Tran* can be considered as variants of *Seq*, *Seq_Dur*, *Seq_Tran*, and *Seq_Dur_Tran* that consider location popularity. The results in Figure 4.3 show that these variants achieve better results than the original ones. However, the differences are not significant (Seq_Idf vs. Seq: $p=0.22$; Seq_Idf_Dur vs. Seq_Dur: $p=0.17$; Seq_Idf_Tran vs. Seq_Tran: $p=0.09$; Seq_Idf_Dur_Tran vs. Seq_Dur_Tran: $p=0.27$). The non-significant results might be due to the small amount of user profiles, which cannot accurately reflect the actual popularity of locations.

Considering duration at locations: *Seq_Dur*, *Seq_Idf_Dur*, *Seq_Dur_Tran* and *Seq_Idf_Dur_Tran* can be considered as variants of *Seq*, *Seq_Idf*, *Seq_Tran*, and *Seq_Idf_Tran* that consider duration at locations. Figure 4.3 shows that these variants achieve better results than the original ones. However, the differences are not significant (Seq_Dur vs. Seq: $p=0.26$; Seq_Idf_Dur vs. Seq_Idf: $p=0.12$; Seq_Dur_Tran vs. Seq_Tran: $p=0.09$; Seq_Idf_Dur_Tran vs. Seq_Idf_Tran: $p=0.27$). The non-significant results might be due to the way we extracted user profiles from raw location histories as a sequence of stops and moves. Recall that a stop is defined when a user has stayed in a location over a duration threshold. The use of duration threshold might already reduce the variance of duration at locations, which leads to a very close similarity value on the aspect of duration at locations.

Considering transit time between locations: *Seq_Tran*, *Seq_Idf_Tran*, *Seq_Dur_Tran* and *Seq_Idf_Dur_Tran* can be considered as variants of *Seq*, *Seq_Idf*, *Seq_Dur*, and *Seq_Idf_Dur* that consider transit time between locations. Figure 4.3 shows that these variants achieve better results than the original ones, and the differences are significant, except Seq_Tran vs. Seq (Seq_Tran vs. Seq: $p=0.20$; Seq_Idf_Tran vs. Seq_Idf: $p=0.03$; Seq_Dur_Tran vs. Seq_Dur: $p=0.02$; Seq_Idf_Dur_Tran vs. Seq_Idf_Dur: $p=0.047$).

In order to have a clearer comparison of the effectiveness of the above properties, we visualize the differences between the original similarity measures with their variants. Figure 4.4 shows the results. In this figure, the effectiveness of each property was computed as the averaged differences between the original similarity measures and their variants that consider the property. For example, the effectiveness of "transit between locations" was measured as the average of the following differences: *Seq* vs. *Seq_Tran*, *Seq_Idf* vs. *Seq_Idf_Tran*, *Seq_Dur* vs. *Seq_Dur_Tran*, and *Seq_Idf_Dur* vs. *Seq_Idf_Dur_Tran.*

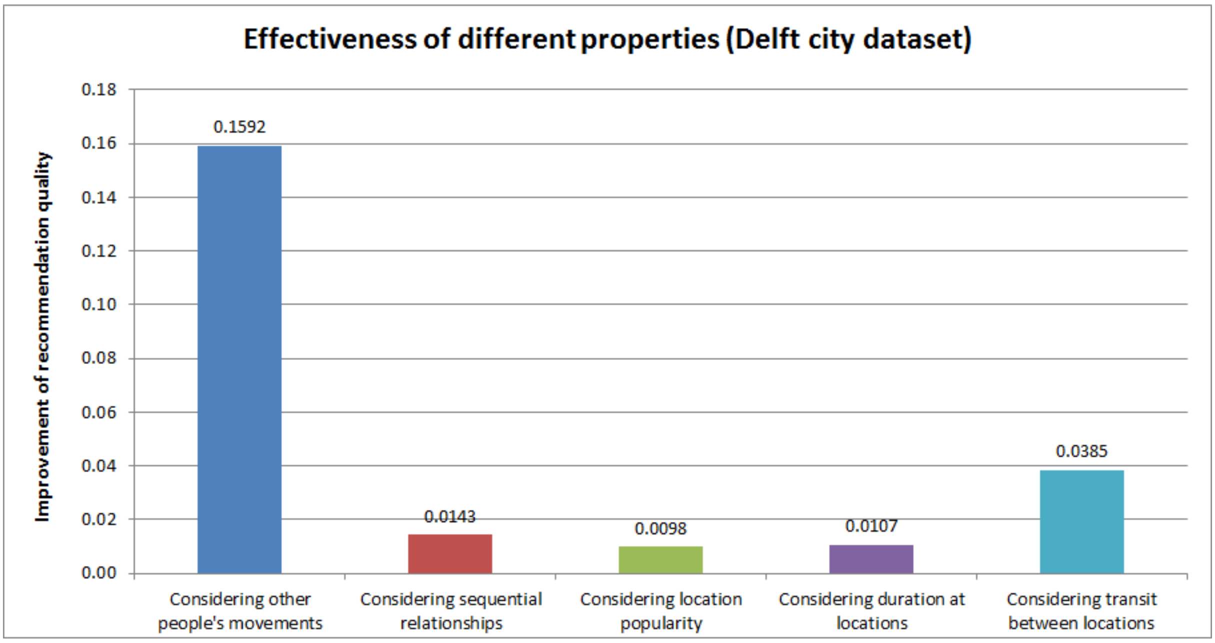


Figure 4.4 Comparison of the effectiveness of different properties (Delft city dataset)

Comparison of different properties: Figure 4.4 shows that all the properties have a positive effect on the recommendation quality. However, their effects differ greatly. Considering other people's movements achieves the biggest improvement, followed by transit time between locations, sequential relationships, duration at locations, and finally location popularity. Please refer to Section 4.5.4 (Summary of the results and discussions) for discussions on this aspect.

2) Evaluation with the Vienna zoo dataset

The Vienna zoo dataset was collected in the Vienna zoo (Tiergarten Schönbrunn, Austria), which is a restricted area, and very similar to an indoor scenario. Figure 4.5 depicts the comparison of all the recommendation methods, which used different similarity measures.

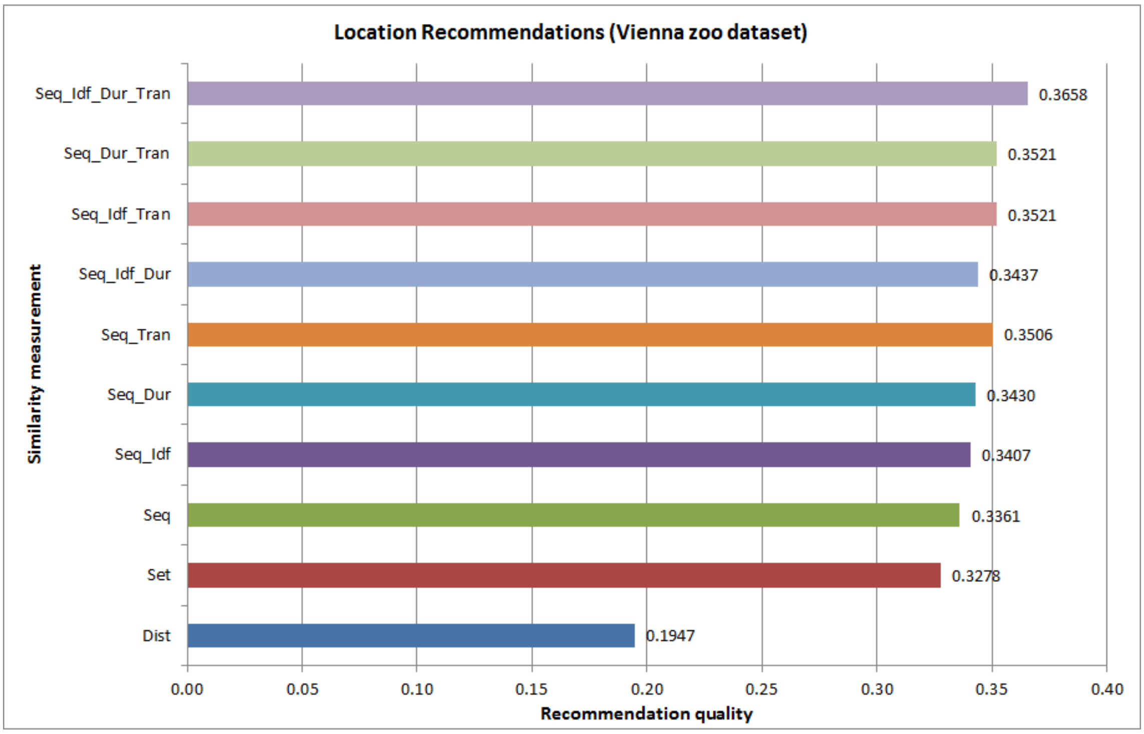


Figure 4.5 Comparison of the recommendation quality among different similarity measures (Vienna zoo dataset)

Similar to the results of the Delft city dataset, the proposed user similarity measure which considers the aspects of sequence relationships of locations visited, location popularity, duration at locations, and transit time between locations achieves the best recommendation quality. In the following, we investigate each aspect/property in detail.

Considering other people's movements: *Set* and *Seq* can be considered as improvements of *Dist* by considering other people's movements. Figure 4.5 shows that both *Set* and *Seq* achieve significant better recommendation quality than *Dist* ($p<0.001$), with improvements of 0.13 and 0.14 respectively. Therefore, considering other people's movements can help to improve recommendation quality.

Considering the orders in which locations are visited (sequential relationships): *Seq* can be considered as an improvement of *Set* by considering sequential relationships. Figure 4.5 shows that *Seq* achieves better results than *Set*. However, the difference is not significant ($p=0.16$).

Considering location popularity: *Seq_Idf*, *Seq_Idf_Dur*, *Seq_Idf_Tran* and *Seq_Idf_Dur_Tran* can be considered as variants of *Seq*, *Seq_Dur*, *Seq_Tran*, and *Seq_Dur_Tran* that consider location popularity.

Figure 4.5 demonstrates that these variants achieve better results than the original ones. However, the differences are not significant, except Seq_Idf_Dur_Tran vs. Seq_Dur_Tran (Seq_Idf vs. Seq: *p=0.48*; Seq_Idf_Dur vs. Seq_Dur: *p=0.26*; Seq_Idf_Tran vs. Seq_Tran: *p=0.47*; Seq_Idf_Dur_Tran vs. Seq_Dur_Tran: *p=0.006*).

Considering duration at locations: *Seq_Dur*, *Seq_Idf_Dur*, *Seq_Dur_Tran* and *Seq_Idf_Dur_Tran* can be considered as variants of *Seq*, *Seq_Idf*, *Seq_Tran*, and *Seq_Idf_Tran* that consider duration at various locations. Figure 4.5 shows that these variants achieve better results than the original ones. However, the differences are not significant (Seq_Dur vs. Seq: *p=0.14*; Seq_Idf_Dur vs. Seq_Idf: *p=0.13*; Seq_Dur_Tran vs. Seq_Tran: *p=0.47*; Seq_Idf_Dur_Tran vs. Seq_Idf_Tran: *p=0.27*).

Considering transit time between locations: *Seq_Tran*, *Seq_Idf_Tran*, *Seq_Dur_Tran* and *Seq_Idf_Dur_Tran* can be considered as variants of *Seq*, *Seq_Idf*, *Seq_Dur*, and *Seq_Idf_Dur* that consider transit time between locations. Figure 4.5 demonstrates that these variants achieve better results than the original ones, and the differences are significant, except Seq_Dur_Tran vs. Seq_Dur (Seq_Tran vs. Seq: *p=0.03*; Seq_Idf_Tran vs. Seq_Idf: *p=0.04*; Seq_Dur_Tran vs. Seq_Dur: *p=0.07*; Seq_Idf_Dur_Tran vs. Seq_Idf_Dur: *p=0.02*).

In order to have a clearer comparison of the effectiveness of the above properties, we also visualize the differences between the original similarity measures with their variants in Figure 4.6.

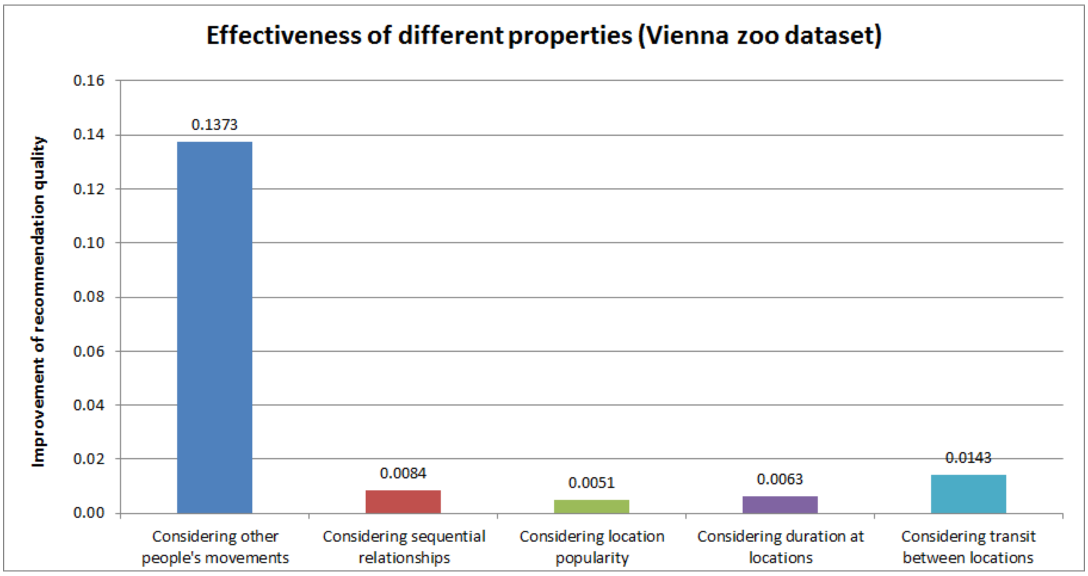


Figure 4.6 Comparison of the effectiveness of different properties (Vienna zoo dataset)

Comparison of different properties: Figure 4.6 shows that all the properties have a positive effect on the recommendation quality. However, their effects differ greatly. Similar to the results of the Delft city dataset, considering other people's movements achieves the biggest improvement, followed by transit time between locations, sequential relationships, duration at locations, and finally location popularity.

3) Evaluation with the Flickr dataset

Compared to the above datasets collected by GPS loggers, the Flickr dataset was constructed from the photos Flickr users uploaded for the city of Vienna during 01.2007 – 01.2011. The application scenario can be considered as a bigger urban scenario, compared to the Delft city center. Please note that, due to the missing duration information in the Flickr dataset, we only compared the *Dist*, *Set*, *Seq*, *Seq_Idf*, *Seq_Tran*, *Seq_Idf_Tran* methods for this dataset. Figure 4.7 depicts the results.

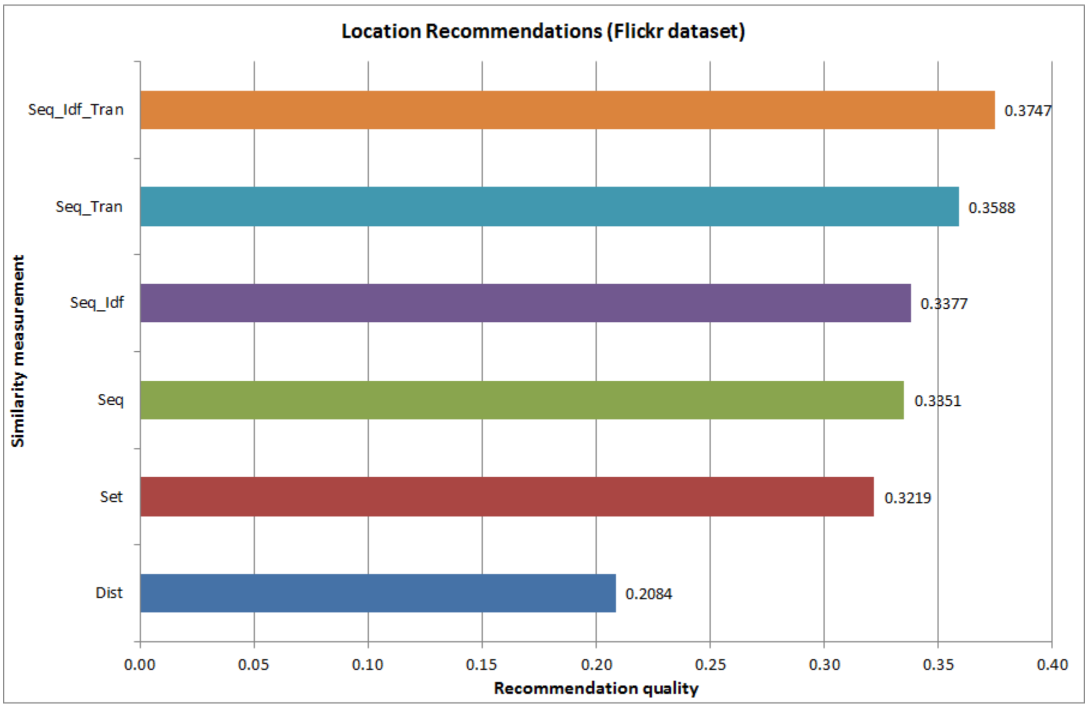


Figure 4.7 Comparison of the recommendation quality among different similarity measures (Flickr dataset)

Similar to the results of the other two datasets, the proposed user similarity measure considering all the proposed aspects achieves the best recommendation quality in the Flickr dataset. In the following, we investigate each aspect/property in detail.

Considering other people's movements: *Set* and *Seq* can be considered as improvements of *Dist* by considering other people's movements. Figure 4.5 shows that both *Set* and *Seq* achieve significant better recommendation quality than *Dist* ($p<0.001$), with improvements of 0.11 and 0.13 respectively. Therefore, considering other people's movements can help to improve recommendation quality.

Considering the orders in which locations are visited (sequential relationships): *Seq* can be considered as an improvement of *Set* by considering sequential relationships. Figure 4.5 shows that *Seq* achieves better results than *Set*. However, the difference is not significant ($p=0.18$).

Considering location popularity: *Seq_Idf* and *Seq_Idf_Tran* can be considered as variants of *Seq* and *Seq_Tran* that consider location popularity. The results in Figure 4.7 show that these variants achieve better results than the original ones. However, the differences are not significant, (Seq_Idf vs. Seq: $p=0.18$; Seq_Idf_Tran vs. Seq_Tran: $p=0.41$).

Considering transit time between locations: *Seq_Tran* and *Seq_Idf_Tran* can be considered as variants of *Seq* and *Seq_Idf* that consider transit time between locations. Figure 4.7 shows that these variants achieve better results than the original ones. However, different from the other two datasets, the improvements are not significant (Seq_Tran vs. Seq: $p=0.22$; Seq_Idf_Tran vs. Seq_Idf: $p=0.10$). Our similarity measure based on transit time between locations did not work well on the Flickr dataset, as users might have many other activities during the transit, which were not represented in the Flickr dataset.

In order to have a clearer comparison of the effectiveness of the above properties, we also visualize the differences between the original similarity measures with their variants. Figure 4.8 shows the results.

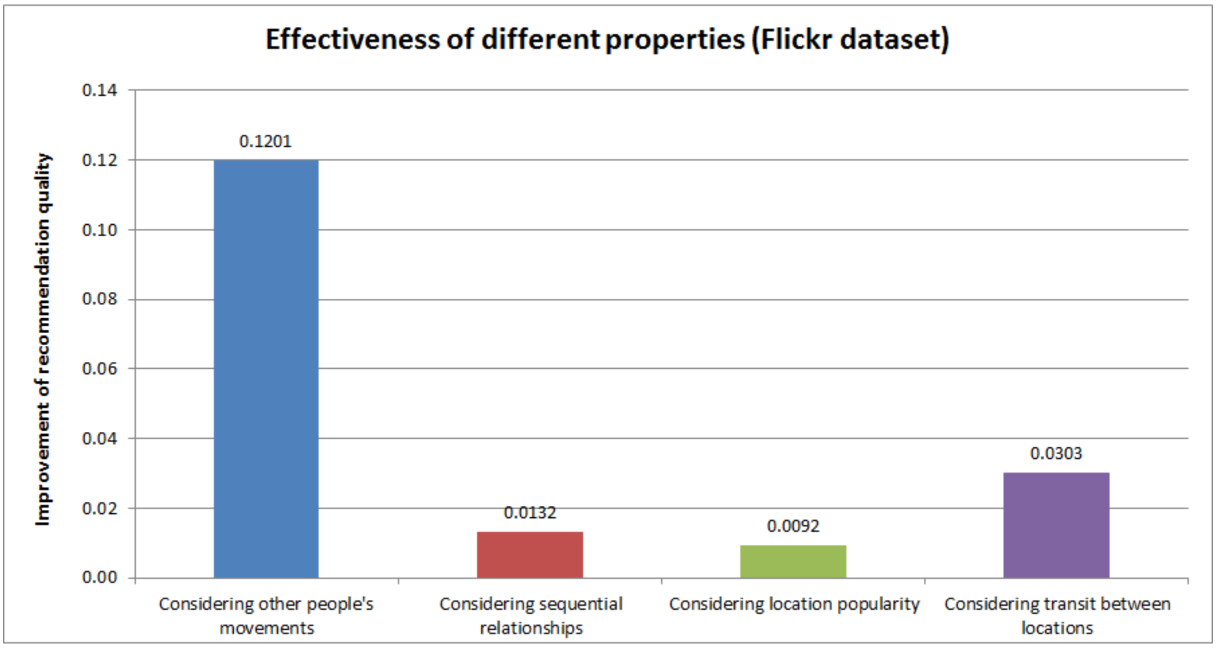


Figure 4.8 Comparison of the effectiveness of different properties (Flickr dataset)

Comparison of different properties: Similar to the results of the other two datasets, considering other people's movements achieves the biggest improvement, followed by transit time between locations, sequential relationships, and finally location popularity.

4.5.4 Summary of the results and discussions

In summary, the results of the three datasets are similar. The main findings of the experiments are as follows:

1) The size of the neighborhood does affect the recommendation quality of the proposed method. After a certain size of neighbors, the recommendation quality becomes stable (Figure 4.1). However, the optimal number of neighbors can be correctly learnt from the training set alone (Figure 4.2). As discussed in Section 4.5.2, these results are probably due to the way we aggregate similar users' movements.
2) All the proposed properties have a positive effect on the recommendation quality. However, their effects differ greatly. Considering other people's movements achieves the biggest

improvement, followed by transit time between locations, sequential relationships, duration at locations, and finally location popularity (Figures 4.4, 4.6 and 4.8).

3) The proposed user similarity measure that considers all the proposed aspects achieves the best recommendation quality, among all the other benchmarking methods (Figures 4.3, 4.5 and 4.7).

In general, these findings are consistent with what we expected. The big improvement when considering other people's movements confirms the findings of Bohnert (2010), which showed that other users' movements are better predictors of the current user's movements than the distances between locations. The results also confirm the findings of Wexelblat (1999) and Zheng *et al.* (2011), and suggest that experiences from past users (especially those users similar to the current user) can help the current user to solve his/her own problems efficiently, e.g., choosing where to visit next.

The interesting result is on the aspect of transit time between locations. It has a greater impact than sequential relationships, duration at locations, and location popularity. This might be explained by the way we extracted user profiles from raw location histories, and the way we made recommendations. Please recall that we used the concepts of stops and moves, and a stop is defined when a user has stayed in a location/place over a duration threshold. With this, we might miss a user's visits to some places/locations whose duration is too short to be defined as stops. To some extent, the similarity measure on the aspect of transit time between locations can help to discover this situation, and therefore, it can provide recommendations that are more appropriate. However, this similarity measure can be also improved by mapping the similarity value to a continuous scale between 1 and 0, or by considering more aspects of transit between locations, e.g., transportation modes and short occurrences at other locations during transit. We expect that having a more comprehensive measure on these aspects will further improve the recommendation quality.

Another interesting result is that considering duration at locations did not bring a significant improvement to the recommendation quality. This might be again explained by the way we extracted user profiles from raw location histories as a sequence of stops and moves. The use of duration threshold for defining a stop might already reduce the variance of duration at locations, which leads to a very close similarity value on the aspect of duration at locations. Therefore, considering duration at locations does not lead to a big improvement of recommendation quality.

It is also interesting to see that for all the three datasets, the orders of the effectiveness of different properties are the same: Considering other people's movements achieves the biggest improvement, followed by transit time between locations, sequential relationships, duration at locations, and finally location popularity. However, conclusions on this aspect should be made carefully, as the differences among the last four properties are very small, and therefore, the relative orders of these four properties might be slightly changed with different datasets.

To sum up, these experiments demonstrate that considering other people's movements, sequence relationships of locations visited, location popularity, duration at locations, and transit time between locations contributes to the improvement of recommendation quality. These experiments also confirmed that the proposed user similarity measure considering all the above aspects achieves the best recommendation quality, among all the other benchmarking methods.

It is important to note that the current evaluation (and also the one in Chapter 5) used the DTF-SMoT method to identify a sequence of stops and moves from each trajectory, which requires a set of candidate stops and their boundaries (polygons) as inputs. In the current evaluation, these inputs were defined by carefully studying the layouts of the scenarios and the accuracy of the location-acquisition technologies. As mentioned in Chapter 3, definition of these inputs can be improved by involving domain experts, such as tourism experts, behavior experts, and experts in zoo management. We expect that with the involvement of domain experts in defining these candidate stops and their boundaries, we can have a more meaningful abstraction of the location histories, and therefore, the proposed recommendation methods will generate better results than what we have in the current evaluation.

4.6 Summary and conclusions

This chapter presented a CF method for deriving personalized location recommendations from a large number of users' location histories. Specifically, we investigated how other users' interests in various locations (reflected by their visits to these locations and the duration of these visits) and motion behavior of visiting such locations, as extracted from their location histories (as in Chapter 3), can be utilized to provide the current user with personalized location recommendations. We explored a novel user similarity measure by considering the sequence property of movement (i.e., the order in which locations are visited), location popularity, duration at locations and transit time between locations. We then employed this user similarity measure to identify users who are similar to the current user, and aggregated the “opinions” (i.e., movements) of these similar users to generate location recommendations for the current user. The proposed CF derives personalized recommendations from the non-intrusive observations of users (i.e., location histories) and does not require users’ explicit inputs, and therefore, it is very suitable for LBS applications, as LBS users are often involved in many other tasks and activities during using their mobile devices.

We evaluated the proposed method for personalized location recommendations, against some benchmarking methods. Three real-world location history datasets were used for the evaluation: GPS trajectories at Delft city center, GPS trajectories at Vienna zoo, and trajectories constructed from Flickr photos uploaded for the city of Vienna. These datasets reflected different scales of application scenarios and consisted of different types of location histories. The results of the evaluation on the three datasets are similar. The evaluation shows that considering other people’s movements, sequence relationships of locations visited, location popularity, duration at locations, and transit time between locations contributes to the improvement of recommendation quality. Among them, considering other people’s movements achieves the biggest improvement. The results are also consistent with what we expected: experiences from past users (especially similar users) can help current users solve their own problems efficiently. These experiments also confirm that the proposed user similarity measure considering all the above aspects achieves the best recommendation quality, among all the other benchmarking methods.

Although the proposed method performs considerably better than the benchmarking methods, the actual performance is modest. These results might be explained by the relatively small sizes of our datasets, as well as a lack of considering contextual information in the recommendation process. We will address the latter issue in the next chapter (Chapter 5). Specifically, the non-contextual CF method proposed in this chapter will be enriched with contextual information to provide users with personalized and context-aware location recommendations. We expect that the recommendation quality will be further improved through the integration of contextual information, such as weather and companion.

5 Improving Location Recommendations through Context-awareness

Chapter 4 presented a non-contextual CF method for deriving personalized location recommendations from a large number of users' location histories. This chapter aims to improve the non-contextual method by integrating contextual information such as weather and companion (e.g., alone or with others). With this, location recommendations matching users' interests and context can be provided in LBS.

There was research focusing on deriving recommendation from location histories or interaction histories in LBS (Takeuchi and Sugimoto 2006, Bohnert 2010, Zheng *et al.* 2011). However, as mentioned in Chapter 2, most of the research only employed the current location as contextual factor, and did not consider contextual factors which are also relevant for generating recommendations, e.g., weather, companion (with whom), and weekend/weekday.

This chapter is structured as follows. Section 5.1 analyzes the key issues of incorporating contextual information into CF in LBS applications. Sections 5.2-5.4 address these key issues. Section 5.2 explores a methodology for identifying relevant context parameters. A novel context similarity measure is proposed in Section 5.3. Section 5.4 investigates ways to integrate the similarity measure into the CF process for making context-aware location recommendations. We evaluate and discuss the proposed methods in Section 5.5. Finally, we summarize the chapter in Section 5.6.

5.1 Key issues of incorporating contextual information into CF

Context-aware collaborative filtering (CaCF) aggregates what similar users chose in similar contexts for recommendations. Several key issues have to be considered when providing CaCF in LBS:

identifying relevant context parameters, measuring context similarity, and incorporating contextual information into the CF process.

1) Identifying relevant context parameters: In Section 3.1, we have developed a model of user profiles, which can be used to represent users' visiting information as extracted from location histories. In order to provide context-aware recommendations, user profiles should be annotated with contextual information, i.e., information about the situation in which users' movement happens. A context (situation) can be characterized by a set of context parameters/dimensions. Not all the context parameters are relevant for generating recommendations. In order to annotate user profiles with context, a key question has to be answered: Which context parameters are relevant and thus needed to be modeled? Many researchers chose some features of the world as context parameters from their own views (e.g., Adomavicius *et al.* (2005), and Panniello and Gorgoglione (2012)). What is missing, however, is a method of identifying relevant context parameters for CaCF in LBS[28].
2) Measuring context similarity: In general, movements in contexts/situations similar to the context of the current user (who asks for recommendations) are more useful for making location recommendations than those happening in dissimilar contexts. Therefore, similarity measures between different contexts/situations should be developed.
3) Incorporating contextual information into the CF process: Adomavicius and Tuzhilin (2011) proposed three approaches to incorporate contextual information into CF: (a) contextual pre-filtering (contextualization of recommendation input): filter out irrelevant ratings (i.e., trajectories in our case) before using the non-contextual CF method; (b) contextual post-filtering (contextualization of recommendation output): use the non-contextual CF method, and then filter the results with contextual information; (c) contextual modeling (contextualization of recommendation algorithms): use contextual information directly inside the recommendation process. Currently, the approaches have not been applied to provide CaCF in LBS. How these approaches can be combined with the other key issues to provide CaCF in LBS should be carefully investigated.

[28] As mentioned in Chapter 1, one exception was given by Keßler (2010). His approach required that the ranking result for each context is available before the identification of relevant context parameters. However, this assumption might be impractical for many LBS applications.

5.2 Identification of relevant context parameters

As mentioned before, contextual user profiles are important for context-aware recommendations. For annotating user profiles with contextual information, we have to answer the question: Which context parameters are relevant and thus needed to be modeled?

We adopt the context definition given by Dey (2001, p. 5): "Context is any information that can be used to characterize the situation of an entity. An entity is a person, place or object that is considered relevant to their interaction between a user and an application." We also adopt a representational view on context proposed by Dourish (2004), as this view is much simpler and more computationally feasible for many applications (Baltrunas 2011). This representational view assumes that context acts as a set of conditions under which an activity (i.e., interaction) occurs, and it could be modeled using a set of observable attributes. These attributes can be considered as context parameters/dimensions, which can help to differentiate/recognize different context (situations). We also understand that something is context (parameter) only if users' decision-making (e.g., choosing which places/locations to visit), interaction with the system, or the behavior of the system depends on it, otherwise it is just a feature of the world (Winograd 2001, Huang and Gartner 2009). For example, the temperature of the room is a relevant context parameter only if the adaptation of the interaction between human and the current system depends on it (or the behavior of the system depends on it, e.g., when the temperature is higher than 30°C, start the air-conditioner), but otherwise it is a feature of the world.

Based on this understanding, a statistical-test-based method to identify context parameters that are relevant for the recommendation tasks is designed. We assume that a preliminary set of context parameters have been identified from the literature, or by domain experts or brainstorming, and data from users are collected in different situations characterized by this preliminary set.

In the following, we describe how the set of relevant context parameters can be extracted by refining the preliminary set according to the collected dataset. The basic strategy of refining is to analyze how some key aspects/characteristics (e.g., the number of locations/places visited) of users' movements differ with different values of each context parameter in the preliminary set. If context parameter *c1* has n values, and the differences of the key aspects of visits are significant among these *n* values, then the current context parameter *c1* is relevant and thus needed to be modeled, otherwise it is irrelevant. Statistical approaches such as independent T-test, analysis of variance (ANOVA), Mann-Whitney Test, and Kruskal-Wallis Test can be employed to test the significance of differences[29]. For example, if context parameter "weather" has two values "sunny" and "rainy", and the difference between the key aspects of visits (e.g., the number of places visited) in "sunny" and the key aspects of visits in "rainy" is significant (i.e., "people behave differently in different weather condition"), then "weather" is relevant and thus needed to be modeled for CaCF, otherwise it is irrelevant.

It is necessary to note that we do not need to consider the current geographic position as a relevant context parameter when annotating user profiles (as extracted from location histories) with context. The reason is that users' current position is already stored in their trajectories. When recommending a location/place for the current user, his/her current position (location) is used to select places matching his interests and context (see step 1 of all the methods in Section 5.4).

5.3 Context similarity measure

In general, movements happening in a context similar to the current one are more useful for making recommendations for the current user. In the following, we explore a statistics-based approach for measuring similarity between different contexts (situations).

With the method proposed in Section 5.2, relevant context parameters can be identified. By varying values for each parameter, all possible situations can be identified. In the following, we propose an approach to measure the similarity between any two situations.

We assume that if visits in a situation (e.g., A) are similar to visits in another situation (e.g., B), these two situations (contexts) can be considered as similar. Please note that, for "visits", we do not mean each individual visit, but rather an aggregation of all the visits happening in the situation. Based on this assumption, we measure the similarity between any two contexts (situations) with the following

[29] As suggested by many statistic textbooks, e.g., Field (2013), the former two tests are for data with a normal distribution, while the latter two are non-parametric tests and meant for non-normal data.

two steps.

1) The profile of each context (situation, e.g., *A*) is represented as a vector $V^A = < w_1^A, w_2^A, \ldots, w_n^A >$. Each member of the vector corresponds to the usage of a location/place in this situation, and therefore, *n* is equal to the number of places defined in the application scenario. We use the term frequency-inverse document frequency (TF-IDF) measure to compute the value of each w_i^A. TF-IDF is often used in the field of information retrieval to measure how important a word is to a document in a collection or corpus. The TF-IDF value increases proportionally with the number of times a word appears in the document, but is offset by the frequency of the word in the corpus (Salton and McGill 1986). The latter part helps to control for the fact that some words are generally more common than others. w_i is computed as:

$$w_i^A = TF_i * IDF_i = \frac{N_{A,i}}{N_{A,.}} * log\frac{N_{.,.}}{N_{.,i}} \qquad \text{(Eq. 5.1)}$$

where $N_{A,i}$ is the number of movements in situation *A* which visit the i^{th} location/place, $N_{A,.}$ represents the number of movements in situation *A*, $N_{.,i}$ denotes the number of movements in all situations which visit the i^{th} location/place, and $N_{.,.}$ is the total number of movements in all situations. The first part of the Eq. 5.1 denotes how often the i^{th} place is visited in situation *A*, while the second part measures whether the place is commonly or rarely visited across all movements.

The profile of a context can be considered as an aggregated view of usage of different locations in this context (situation), which can be used to characterize the situation. In fact, the TF-IDF measure is also often used in content-based recommendation for representing the profile of an item (Adomavicius and Tuzhilin 2005).

2) The similarity between two contexts (situations) can be then computed by the cosine similarity measure. Cosine similarity is often used for measuring the similarity between objects that are represented as vectors, and it measures the cosine of the angle between these two vectors (Singhal 2001). Two vectors with the same orientation have a cosine similarity of 1, while two vectors at 90° have a similarity of 0. Therefore, the similarity between two contexts (situations) can be measured as the cosine similarity between their corresponding profile vectors.

$$Conx\_Sim(A,B) = cos(\theta) = \frac{V_A \cdot V_B}{\|V_A\|\|V_B\|} = \frac{\sum_{i=1}^{n} w_i^A \times w_i^B}{\sqrt{\sum_{i=1}^{n}(w_i^A)^2} \times \sqrt{\sum_{i=1}^{n}(w_i^B)^2}} \quad \text{(Eq. 5.2)}$$

With these two steps, similarity between any two situations can be calculated.

5.4 Context-aware location recommendation

As mentioned in Section 5.1, Adomavicius and Tuzhilin (2011) proposed that contextual information can be incorporated into CF by contextual pre-filtering, contextual post-filtering and contextual modeling. Other research has applied this classification for making movie recommendation and product recommendations (Adomavicius *et al.* 2005, Panniello *et al.* 2009, Panniello and Gorgoglione 2012). However, this classification has not been applied to location recommendations in LBS. In this section, we apply this classification, and develop several methods for deriving context-aware location recommendations from location histories.

5.4.1 Contextual pre-filtering

The basic idea of contextual pre-filtering approaches is to filter out irrelevant movements before using a non-contextual CF method, e.g., the one proposed in Chapter 4. Based on this idea, we develop the following contextual pre-filtering approach (in short, CaCF_Pre).

Assume that the current user *u* has visited a set of locations/places. Currently he/she is at the location *p*, and asking "which place to visit next?". The steps of the CaCF_Pre method are designed as follows:

1) Identifying users (movements) whose next location after visiting *p* has not been visited by the current user;
2) Filtering out users (movements) whose context similarities with the current user do not exceed a threshold δ. Context similarity is measured using the method proposed in Section 5.3;
3) For the results of step 2, identify the N users who are most similar to the current user, using

the user similarity measure in Eq. 4.9 (i.e., Seq_Idf_Dur_Tran);

4) For the N most similar users, aggregating their next locations after visiting *p* (weighted by the user similarity values);
5) Selecting the location with the highest predicted value, and recommending it to the current user.

The above algorithm consists of two parameters: δ – the context similarity threshold, and N – the number of similar users (i.e., neighborhood size). We investigate the sensitivity of these parameters in Section 5.5.

5.4.2 Contextual post-filtering

Compared to contextual pre-filtering CF approaches which filter out irrelevant movements before using a non-contextual CF method, contextual post-filtering approaches first use the non-contextual CF, and then adjust the results according to contextual information. Based on this idea, we develop the following contextual post-filtering approach (in short, CaCF_Post).

Assume that the current user *u* has visited a set of locations. Currently he/she is at the location *p*, and asking “which place to visit next?”. The steps of the CaCF_Post method are designed as follows:

1) Identifying users whose next location after visiting *p* has not been visited by the current user;
2) Identifying the N users who are most similar to the current user, using the user similarity measure in Eq.4.9 (i.e., Seq_Idf_Dur_Tran);
3) For the N most similar users, aggregating their next locations after visiting *p* (weighted by the user similarity values); The results of this step are a set of candidate locations/places and their corresponding predicted values;
4) For each candidate location from the results of step 3, computing its visit probability for the current context. The visit probability of a location p is computed as the percentage of neighbors (i.e., similar users) who visited the location/place in similar contexts (i.e., contexts whose similarity value with the current context c_1 is bigger than a threshold δ).

$$Visit\_Prob\ (p) = \frac{|\{o|\ o \epsilon\ neighbors \wedge visit\ (o,p,c_2) \wedge \mathrm{Conx\_Sim}(c_1,c_2) > \delta\}|}{|neighbors|} \quad \text{(Eq. 5.3)}$$

where the denominator denotes the number of neighbors (=N), and the numerator represents the number of neighbors who visited the location/place in similar contexts.

The final predicted value for each candidate location is computed as:

$$Pred\_Value\_Conx\ (p) = pred\_v(p) \times Visit\_Prob(p) \qquad \text{(Eq. 5.4)}$$

Where $pred\_v(p)$ is the predicted value computed from step 3.

5) Selecting the location with the highest predicted value, and recommending it to the current user.

The above algorithm consists of two parameters: δ – the context similarity threshold, and N – the number of similar users (i.e., neighborhood size). We investigate the sensitivity of these parameters in Section 5.5.

5.4.3 Contextual modeling

Compared to the above two approaches, contextual modeling approaches use contextual information directly inside the CF process. Based on this idea, we develop the following contextual modeling approach (in short, CaCF_Mdl).

Assume that the current user *u* has visited a set of locations/places. Currently he/she is at the location *p*, and asking "which place to visit next?". The steps of the CaCF_Mdl method are designed as follows:

1) Identifying users whose next location after visiting *p* has not been visited by the current user;
2) Identifying the top N users whose movements are most useful for making recommendations for the current user. The utility of a user b's movement to the current user is measured as a combination of user similarity and context similarity.

$$Utility(b, u) = \mathrm{SIM}_{user}(b, u) * \mathrm{SIM}_{conx}(b, u) \qquad \text{(Eq. 5.5)}$$

where $\mathrm{SIM}_{user}(b, u)$ measures the user similarity between b and u, using the user similarity measure in Eq. 4.9 (i.e., Seq_Idf_Dur_Tran). $\mathrm{SIM}_{conx}(b, u)$ denotes the context similarity between the contexts of the movements of b and u, using the context similarity measure in Eq. 5.2.

3) For these N users, aggregating their next locations after visiting *p* (weighted by the utility values);
4) Selecting the location with the highest predicted value, and recommending it to the current user.

The above algorithm consists of one parameter N – the neighborhood size. We investigate the sensitivity of this parameter in Section 5.5.

5.4.4 Comparisons

When comparing the above methods, one might discover that they differ in the aspect of when user similarity measure and context similarity measure are introduced. CaCF_Pre uses context similarity measure and then employs user similarity measure. CaCF_Post uses them in a complete inverse way. Compared to CaCF_Pre and CaCF_Post, CaCF_Mdl combines both measures to generate a utility measurement to measure the usefulness of a movement for making recommendations for the current user.

5.5 Evaluation and discussions

This section evaluates the above methods for context-aware location recommendations with the two real-world location history datasets introduced in Chapter 3 (Delft city dataset and Vienna zoo dataset). In Section 5.5.1, we describe how we processed these datasets. Section 5.5.2 employs the proposed method in Section 5.2 to identify relevant context parameters. We describe the experimental setting in Section 5.5.3. The evaluation and results are presented and discussed in Section 5.5.4 and Section 5.5.5, and summarized in Section 5.5.6.

5.5.1 Datasets

As mentioned in Section 3.3, the Delft city dataset and the Vienna zoo dataset consisted of movements (recorded as GPS trajectories) and some additional information about each movement, such as weather and companion (with whom). Therefore, these two datasets were employed for the evaluation of the proposed CaCF methods. In the following, we describe the details of both datasets, with a focus on their contextual information.

1) Delft city dataset

In the Delft city dataset, in addition to the movement trajectories, some information about users and their visits was also recorded. Table 5.1 shows the additional information recorded.

Table 5.1 Additional information recorded together with GPS trajectories (Delft city dataset)

Additional information	**Description or available options**
Date	The visit date
Start location	Zuipoort ZP or Phoenix PH
Purpose of visit	Shopping, tourism, leisure, other
With whom	Alone, with partner, with kids, family (kids + partners), other
First visit	Yes/No
Personal data	Gender, age, origin, occupation
Weather	Sunny, cloudy, rainy, rain, windy

In order to alleviate the problem of data sparsity, we processed the above information by aggregating some of the values. Following is the final list of all the additional information we used for the final experiments. Please note that, we did not use personal data (such as gender and age), as they are static attributes of the users.

a) ***marketday: Yes/No***. On Thursday and Saturday, a market is held in Delft city center, which might affect people's visiting behavior. Therefore, we mapped the visit date to either marketday or non-marketday.
b) ***Start_loc: ZP/PH***. This is the location where users started their visits.

c) ***Purpose_of_visit: shopping/not-shopping***. Due to the number of visits labeled as tourism, leisure and others is small, we aggregated visits of these three types as “not-shopping”.

d) ***First_visit: Yes/No.*** This dimension indicates whether this is a user’s first visit to the Delft city center or not.

e) ***With_whom: alone/with kids/with others***. We introduced “with kids” to replace both “with kids” and “family (kids + partners)”.

f) ***Weather: rainy/not rainy***. In the original recorded information, weather was reported as a vector, whose member denoted each dimension. An example is <sunny: no, cloudy: yes, rainy: no, rain: no, windy: yes>. Due to the number of visits happening in many weather conditions is very small, we simplified it as either rainy or not-rainy[30].

The above aspects can be viewed as the preliminary set of contextual dimensions (parameters). In the following sections, we use the method proposed in Section 5.2 to extract context parameters that are relevant for making location recommendations.

Similar to Chapter 4, we used the duration-threshold-free SMoT (DTF-SMoT) as proposed in Section 3.2.1 to extract a sequence of stops and moves from each GPS trajectory. We also removed movements with incomplete contextual information, as well as movements that stopped at less than six locations. In total, we got 114 movements, which were used for the final experiments.

2) Vienna zoo dataset

In the Vienna zoo dataset, the following information was recorded together with GPS trajectories: gender, age, having an annual pass (yes/not), with whom (alone/with kids/with others), and weather (sunny/cloudy, or rainy). For our experiments, we also did not use gender and age, as they are static attributes of the users. In addition, we added a new dimension to indicate whether the visits happed on holidays (weekend and public holidays) or not. Therefore, the final list of all the additional information we used for our final experiments is: holiday (yes or not), annual_pass (yes or not), with_whom (alone/with kids/with others), and weather (rainy/non rainy).

[30] Other classifications of weather conditions are also possible. However, due to the small sizes of visits, we decided to classify weather conditions into rainy and non-rainy. In an intuitive sense, rainy or not might affect users’ visiting behavior for outdoor scenarios.

Similarly, the above aspects can be viewed as the preliminary set of context parameters. In Section 5.5.2, we identify context parameters that are relevant for making location recommendations, using the proposed method in Section 5.2.

Similar to the evaluation in Chapter 4, we used DTF-SMoT as proposed in Section 3.2.1 to extract a sequence of stops and moves from each GPS trajectory. We also removed movements with incomplete contextual information, as well as movements that stopped at less than six locations. In total, we got 165 movements, which were used for the final experiments.

5.5.2 Identifying relevant context parameters

The recorded contextual information (i.e., <marketday, start_loc, purpose_of_visit, first_visit, with_whom, weather> for the Delft city dataset, and <holiday, annual_pass, with_whom, weather> for the Vienna zoo dataset) can be considered as the preliminary set of context parameters. In the following, we apply the method proposed in Section 5.2 to identify relevant context parameters from these preliminary sets.

As we were interested in identifying context parameters that are relevant for making location recommendations, we mainly compared the number of visited locations/places among different situations. In order to test whether the differences among different conditions for each context parameter were significant, we employed a significance test. For each test, we first used Kolmogorov-Smirnov test to investigate whether the data were normally distributed or not. If yes, an independent T-test or an independent one-way ANOVA (Analysis of Variance) was employed, otherwise, non-parametric tests such as Mann-Whitney test or Kruskal-Wallis test were employed. $p < 0.05$ was used to denote statistical significance. Table 5.2 and Table 5.3 show the results of the comparison for both datasets. Each data cell in Table 5.2 and Table 5.3 contains the following information: p-value of significance tests, mean of condition1, mean of condition2, and so on.

Table 5.2 How visits differed among different conditions for each context parameter (Delft city dataset)

Context parameters	**The number of places visited**
'marketday' (Yes/No)	p=0.977 (8.90 vs. 8.83)
'start_loc' (ZP/PH)	p=0.189 (8.56 vs. 9.20)
'purpose_of_visit' (shopping/not shopping)	p=0.682 (8.76 vs. 9.21)
'first_visit' (Yes/No)	p=**0.041 (9.94 vs. 8.69)**
'with_whom' (alone/with kids/with others)	p=**0.043 (7.93 vs. 8.83 vs. 9.35)**
'weather' (rainy/not rainy)	p=0.431 (9 vs. 8.81)

Table 5.3 How visits differed among different conditions for each context parameter (Vienna zoo dataset)

Context parameters	**The number of places visited**
'holiday' (Yes/No)	p=0.712 (11.75 vs. 12.03)
'annual_pass' (Yes/No)	**p<0.001 (10.46 vs. 13.72)**
'with_whom' (alone/with kids/with others)	p=0.382 (11.22 vs. 11.54 vs. 12.62)
'weather' (rainy/not rainy)	**p=0.009 (9.67 vs. 12.30)**

For the Delft city dataset, the significance tests indicated that the numbers of visited places among different "first_visit" conditions were significantly different. Similarly, for different "with_whom" conditions, people also behaved differently (see the bold parts in Table 5.2). Therefore, "first_visit" and "with_whom" were considered as relevant, and taken as the final set of context parameters. Similarly, for the Vienna zoo dataset, "annual_pass" and "weather" were relevant, and therefore taken as the final set of context parameters. In the following section, we provide an evaluation to test the correctness of these decisions.

Please note that, current comparisons on different conditions mainly focused on the number of visited places. We are aware of other possible attributes that can help to characterize travel behavior, such as length of visits and duration of visits. However, as we were interested in identifying context parameters that are relevant for making location recommendations, we mainly compared the number of visited locations/places among different situations. In the following experiments, we show that even with this simple attribute, the results are already promising.

As mentioned in Section 5.5.1, in order to alleviate the problem of data sparsity, we aggregated the original categorical values for some of the initial context parameters in the Delft city dataset, e.g., for 'weather', we aggregated different weather conditions, and classified them into 'rainy' and 'not rainy'. This kind of aggregation might increase or decrease the observed influence of the context parameter, and might lead to different results. However, as the proposed method for identifying relevant context parameters is a statistics-based approach, sometimes, aggregation of the original categories is needed to allow for meaningful significance tests, especially when the datasets for some of the original categories are rather small. Research on Statistics provides some hints on what makes a significance test statistically meaningful (Field 2013), which can be used to decide whether aggregation is necessary or not. Domain knowledge should be also applied during aggregation.

It is also important to note that, the relevant context parameters identified above are valid for the current datasets only. For other datasets collected in different scenarios or even collected in the same scenarios, the list of relevant context parameters might be different. Our research does not aim at identifying a set of context parameters that are generally applicable to the recommendation scenarios. Instead, the goal is to explore a methodology that can help to identify relevant context parameters from each dataset.

5.5.3 Experimental settings

We used the datasets in Section 5.5.1 to evaluate the recommendation performance of different CaCF methods, against the non-contextual method proposed in Chapter 4.

Main objectives. For the experimental evaluation, we were mainly interested in investigating whether including contextual information in CF can improve the recommendation performance or not. In addition, we aimed to evaluate the effectiveness of the proposed methodology (Section 5.2 and Section 5.5.2) in identifying relevant context parameters.

Methods for evaluation: In order to address the above objectives, the following recommendation methods were evaluated in the experiments: the non-contextual method (nonCaCF, Chapter 4), contextual pre-filtering method (CaCF_Pre, Section 5.4.1), contextual post-filtering method (CaCF_Post, Section 5.4.2), and contextual modeling (CaCF_Mdl, Section 5.4.3).

Evaluation metrics. Similar to Chapter 4, the performance of each method was measured as the percentage of successful recommendations, while successful recommendations referred to the case when the recommended location is actually visited immediately by the current user.

Evaluation framework. We used a leave-one-out validation. We made recommendations starting from each user's fourth location, i.e., we did not make recommendations for the first four locations of each visit.

Among all the methods for evaluation, nonCaCF and CaCF_Mdl have one parameter to calibrate, i.e., the number of similar users (neighborhood size) N, while CaCF_Pre and CaCF_Post have one more parameter to calibrate, i.e., the context similarity threshold δ. In the experiments, we first implemented a sensitivity analysis to study the impact of these parameters on recommendation quality. We then evaluated the performance of the proposed CaCF methods when using different sets of context parameters. We were interested in studying whether using the sets of relevant context parameters identified in Section 5.5.2 achieves the best recommendation quality. This evaluation was very useful for testing the effectiveness of the methodology proposed in Section 5.2 (i.e., identifying relevant context parameters). Finally, we compared the CaCF methods with the non-contextual method to study the improvement brought by introducing contextual information. For the last two experiments, we only used the parameter values achieving the best results.

5.5.4 Evaluation results and discussions: Delft city dataset

In this section, we describe the evaluation results of the Delft city dataset, while the results of the Vienna zoo dataset are reported in the next section.

1) Sensitivity analysis

In order to determine the effect of the parameters, we performed some experiments by varying neighborhood sizes and varying context similarity thresholds. For these experiments, we used the set of relevant context parameters identified in Section 5.5.2, i.e., “<first_visit, with_whom>”. The results are shown in Figure 5.1. Please note that Figure 5.1 is generated by keeping the other parameters fixed at their optimum values. By this, our sensitivity analysis focused on all methods' recommendation quality around the highest-ranked configuration.

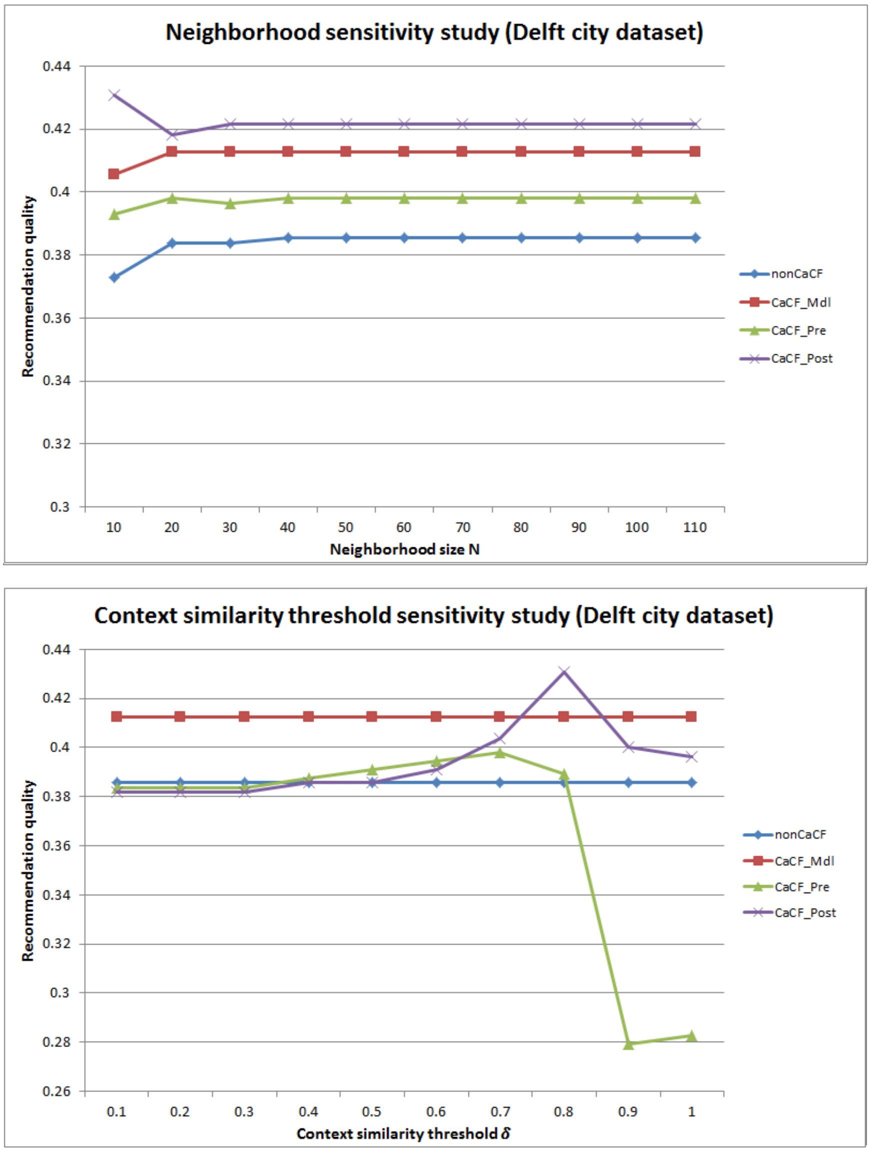


Figure 5.1 Impact of neighborhood size and context similarity threshold on recommendation quality (Delft city dataset)

As can be seen from Figure 5.1, the neighborhood size does affect the recommendation quality of all the methods. Similar to the sensitivity analysis in Chapter 4, after a certain point, the recommendation quality becomes stable for all the methods. This is probably because the recommended location is an aggregated result of similar users' movements, weighted by their similarity values. For each user, only a small group of users have higher similarity values with him/her, and all the other users have a similarity value closer to 0. Therefore, after a certain point, the recommended location remains the same, and the recommendation quality becomes stable.

Concerning the context similarity threshold, the recommendation quality of both CaCF_Pre and CaCF_Post increases when increasing the threshold. However, after a certain point, the quality becomes worse. Please note that, setting the threshold as 1 means that we only use the movements happening in the same context (situation) as the current users. With this, we might not have sufficient movements for making relevant recommendations. Figure 5.1 also shows that the impact effect of the context similarity threshold for CaCF_Pre is bigger than that for CaCF_Post, especially when the threshold is close to 1. This is probably because CaCF_Pre uses the threshold for filtering out movements before the CF process.

For the rest of the experiments in this section, we used a neighborhood of size 40 for nonCaCF, that of 20 for CaCF_Pre, that of 10 for CaCF_Post, and that of 20 for CaCF_Mdl. We also used a context similarity threshold of 0.7 for CaCF_Pre, and that of 0.8 for CaCF_Post.

2) Impact of different sets of context parameters on recommendation quality

This experiment studied whether using the proposed set of context parameters "<first_visit, with_whom>" can achieve the best performance among all the possible sets of context parameters. As we had six preliminary context parameters, in total we had another 62 = $\binom{6}{1} + \binom{6}{2} + \binom{6}{3} + \binom{6}{4} + \binom{6}{5} + \binom{6}{6} - 1$ possible sets of context parameters. Figure 5.2 shows how the recommendation performance of all the CaCF methods changes when using different sets of context parameters.

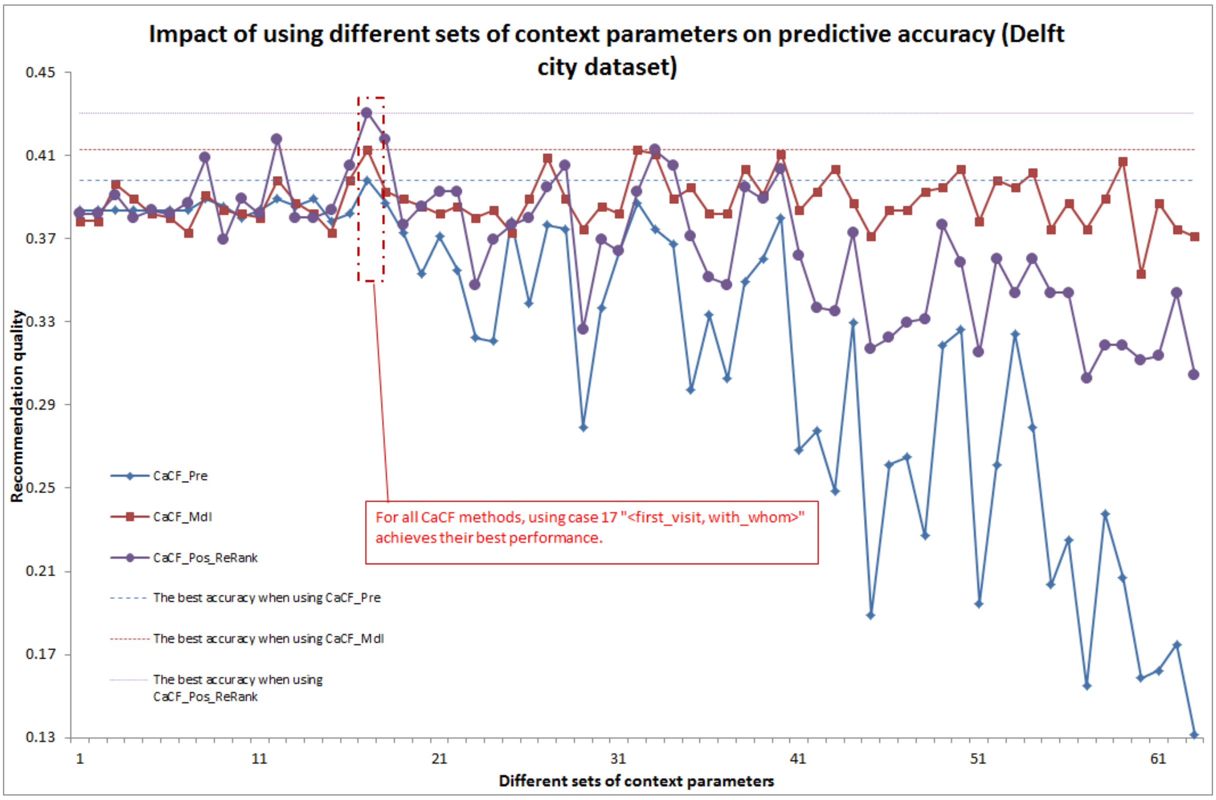


Figure 5.2 The recommendation performance of the proposed CaCF methods changes when using different sets of context parameters (Delft city dataset).

Using different sets of context parameters: Figure 5.2 shows that among all the possible sets of context parameters, all CaCF methods using the proposed set *"<first_visit, with_whom>"* achieve the best recommendation performance. Therefore, the proposed method (in Section 5.2 and Section 5.5.2) to identify relevant context parameters is feasible and useful.

In the meantime, it is also important to note that incorporating more context parameters into the CF process did not mean an improvement of performance. This can be explained by the increasing difficulty of accurately measuring context similarity when using more context parameters, and the increasing demand of data.

3) Comparison of CaCF and nonCaCF methods

In order to experimentally study whether including contextual information in CF can improve the recommendation performance, we compared all the CaCF methods with the non-contextual method (nonCaCF). Figure 5.3 shows the comparison. The proposed set of context parameters *"<first_visit, with_whom>"* was employed. All statistical tests in the following were one-tailed paired t-tests at the significance level $\alpha = 0.05$.

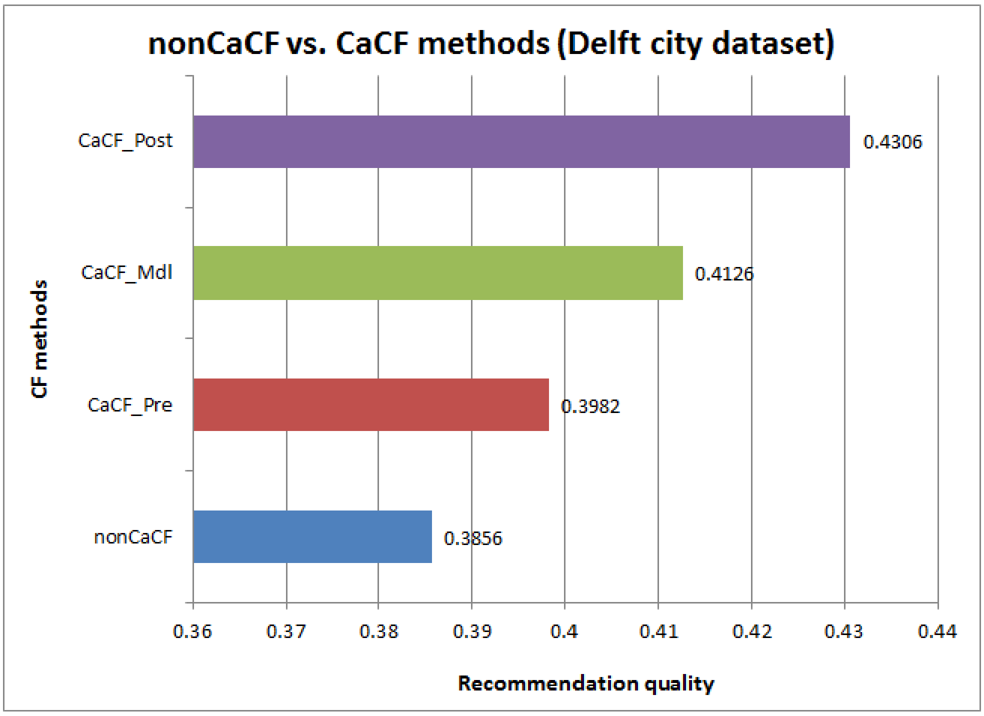


Figure 5.3 Comparison of recommendation quality among the non-contextual method and the CaCF methods (Delft city dataset)

non-contextual CF vs. context-aware CF (CaCF): The recommendation performance of CaCF methods (i.e. CaCF_Pre, CaCF_Mdl, and CaCF_Post) is better than the performance of the non-contextual CF method (i.e. nonCaCF). More specifically, both CaCF_Post and CaCF_Mdl achieve significant better quality than nonCaCF, with improvements of 11.68% and 7.01% respectively (CaCF_Post vs. nonCaCF: *p=0.007*; CaCF_Mdl vs. nonCaCF: *p=0.001*). CaCF_Pre achieves better results than nonCaCF, with a non-significant difference (*p=0.13*).

This is consistent with what we expected: as CaCF methods are aware of the context (situation) the user is in, they might generate recommendations that are more suitable to visit.

Contextual pre-filtering (CaCF_Pre) vs. contextual post-filtering (CaCF_Post) vs. contextual modelling (CaCF_Mdl): Among all the CaCF methods, CaCF_Post performs the best, followed by CaCF_Mdl, finally CaCF_Pre. The difference between CaCF_Post and CaCF_Pre is significant (*p=0.017*). In addition, the difference between CaCF_Post and CaCF_Mdl is close to significant (*p=0.083*). The diverse performance of CaCF methods might be explained by the ways they incorporate contextual information: CaCF_Pre filters out movements that happened in dissimilar situations, which might cause the data sparsity problem; CaCF_Mdl uses context similarity and user similarity to measure the utility of each movement, which makes use of all the movements, but at the same time might introduce some other uncertainties; CaCF_Post generates a set of candidate locations using the non-contextual method, which uses all the movements; the candidate results are then adjusted according to contextual information.

5.5.5 Evaluation results and discussions: Vienna zoo dataset

This section reports on the evaluation results of the Vienna zoo dataset.

1) Sensitivity analysis

Similar to what we did for the Delft city dataset, we performed some experiments by varying neighborhood sizes and varying context similarity thresholds. For these experiments, we used the set of relevant context parameters identified in Section 5.5.2, i.e., *"<annual_pass, weather>"*. The results are shown in Figure 5.4.

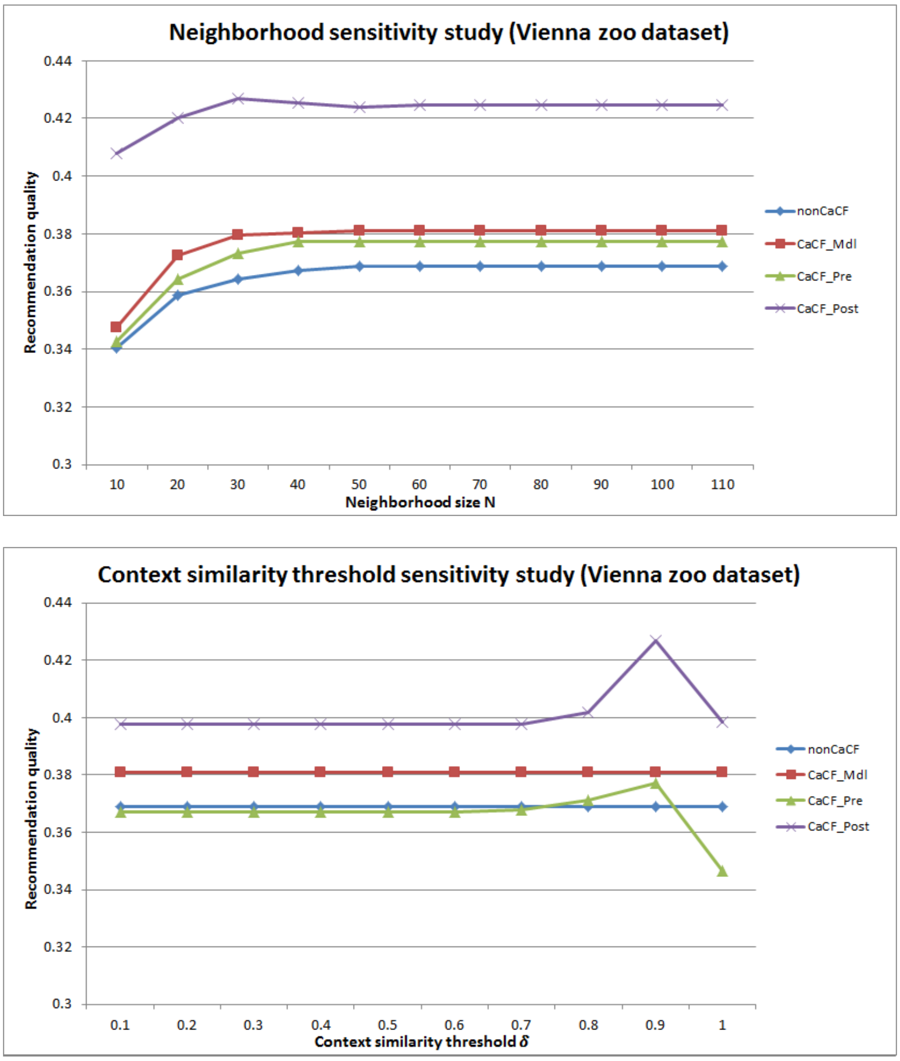


Figure 5.4 Impact of neighborhood size and context similarity threshold on recommendation quality (Vienna zoo dataset)

In general, the results of the Vienna zoo dataset show the same trends as those of the Delft city dataset. In terms of the neighborhood size, after a certain point, the recommendation quality becomes stable for all the methods. Again, this is probably because the recommended location is an aggregated result of similar users' movements, weighted by their similarity values. Concerning the similarity threshold, the recommendation quality of both CaCF_Pre and CaCF_Post increases as the threshold increases. Both methods achieve their peak at 0.9. After that, the quality becomes worse.

Please note that the performance of CaCF_Pre is slightly poorer than nonCaCF. This is because: Compared to nonCaCF, CaCF_Pre filters movements happening in dissimilar contexts, which might cause the data sparsity problem.

For the rest of the experiments in this section, we used a neighborhood of size 50 for nonCaCF, that of 40 for CaCF_Pre, that of 30 for CaCF_Post, and that of 50 for CaCF_Mdl. We also used a context similarity threshold of 0.9 for both CaCF_Pre and CaCF_Post.

2) Impacts of different sets of context parameters on recommendation quality

Similarly, we also studied whether using the proposed set of context parameters *"<annual_pass, weather>"* can achieve the best performance among all the possible sets of context parameters. As we had four preliminary context parameters, in total we had another $14 = \binom{4}{1} + \binom{4}{2} + \binom{4}{3} + \binom{4}{4} - 1$ possible sets of context parameters. Figure 5.5 shows how the recommendation performance of all the CaCF methods changes when using different sets of context parameters.

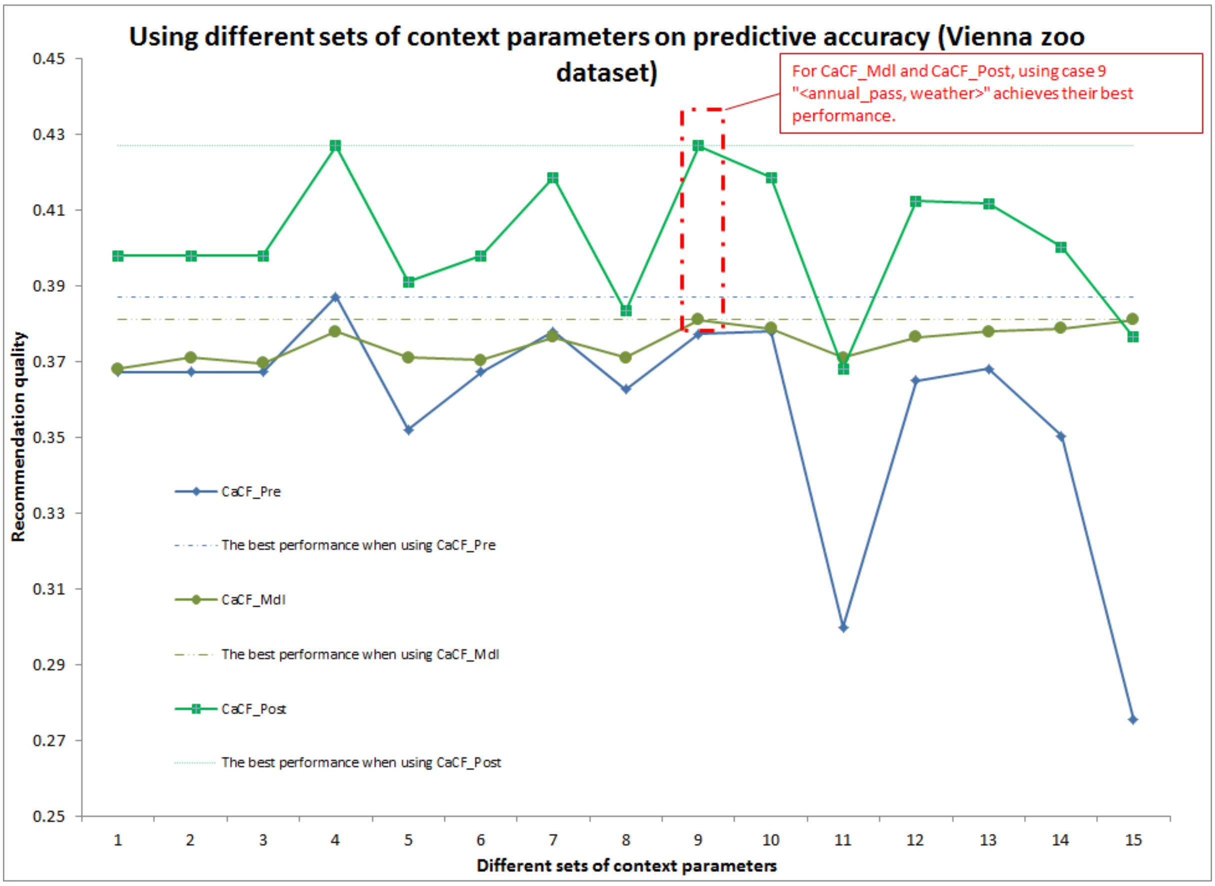


Figure 5.5 The recommendation performance of the proposed CaCF methods changes when using different sets of context parameters (Vienna zoo dataset).

Using different sets of context parameters: Similar to the results of the Delft city dataset, among all the possible sets of context parameters, CaCF_Post and CaCF_Mdl methods using the proposed set *"<annual_pass, weather>"* achieve the best recommendation performance. This is not the case for CaCF_Pre, which achieves the best performance when using "<weather>". However, the performance difference of using "<annual_pass, weather>" and "<weather>" is very small (<0.01).

Figure 5.5 also shows that including more context parameters into CF did not mean an improvement of performance. This can be again explained by the increasing difficulty of accurately measuring context similarity when using more context parameters, and the increasingly demand of data.

3) Comparison of CaCF and nonCaCF methods

Similarly, we compared all the CaCF methods with the non-contextual method (nonCaCF) to experimentally study whether including contextual information in CF can improve the recommendation performance. Figure 5.6 shows the comparison. The proposed set of context parameters *"<annual_pass, weather >"* was employed.

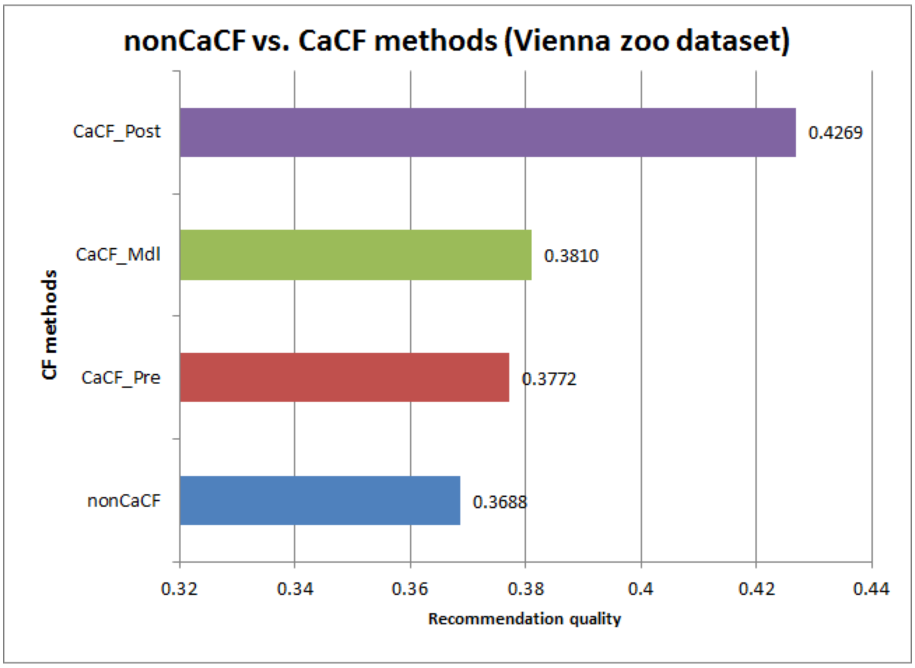


Figure 5.6 Comparison of recommendation quality among the non-contextual method and the CaCF methods (Vienna zoo dataset)

In general, the results of the Vienna zoo dataset are very similar to those of the Delft city dataset.

non-contextual CF vs. context-aware CF (CaCF): The recommendation performance of CaCF methods (i.e. CaCF_Pre, CaCF_Mdl, and CaCF_Post) is better than the performance of the non-contextual CF method (i.e. nonCaCF). More specifically, both CaCF_Post and CaCF_Mdl achieve significant better quality than nonCaCF, with improvements of 15.76% and 3.32% respectively (CaCF_Post vs. nonCaCF: $p<0.001$; CaCF_Mdl vs. nonCaCF: $p=0.002$). CaCF_Pre achieves better results than nonCaCF, with a non-significant difference ($p=0.098$).

Contextual pre-filtering (CaCF_Pre) vs. contextual post-filtering (CaCF_Post) vs. contextual modelling (CaCF_Mdl): Among all the CaCF methods, CaCF_Post performs the best, followed by CaCF_Mdl, finally CaCF_Pre. The difference between CaCF_Post and CaCF_Pre is significant (p<0.001). In addition, the difference between CaCF_Post and CaCF_Mdl is also significant (p=0.003). These results are consistent with the results in the Delft city dataset. Again, the diverse performance of CaCF methods might be explained by the ways they incorporate contextual information.

5.5.6 Summary of the results and discussions

In summary, the results of the two datasets are similar. The main findings of the experiments are as follows:

1) The size of the neighborhood N and the context similarity threshold δ do affect the recommendation quality of all the methods. The trends of these effects in both datasets are similar (Figure 5.1 and Figure 5.4).
2) When including contextual information in the CF process, choosing a suitable set of relevant context parameters is very important and may greatly affect the recommendation performance. Using the proposed set of context parameters, as identified by the proposed method in Section 5.1, achieves the best results (except CaCF_Pre for the Vienna zoo dataset) (Figure 5.2 and Figure 5.5).
3) Among all the CaCF methods, CaCF_Post performs the best, followed by CaCF_Mdl, finally

CaCF_Pre. The difference between CaCF_Post and CaCF_Pre is significant (Figure 5.3 and Figure 5.6).

4) More importantly, the recommendation performance of the CaCF methods (i.e. CaCF_Pre, CaCF_Mdl, and CaCF_Post) is better than the performance of the non-contextual CF method (i.e. nonCaCF). The difference of CaCF_Post and nonCaCF, and that of CaCF_Mdl and nonCaCF are significant.

In general, these findings are consistent with what we expected.

1) We believed that choosing a suitable set of context parameters is very important, and the identification of this set can be achieved by analyzing the travel behavior at different conditions. The results of the experiments confirm this expectation, and show that the proposed methodology is feasible and useful to identify context parameters that are relevant to the recommendation task.

   Please note that the sets of relevant context parameters identified using this methodology are only valid for the current datasets. It does not mean that they are relevant for all other datasets collected in the same scenarios. Our research does not aim to identify a set of context parameters that work in general for the scenarios. Instead, the goal is to explore a methodology that can help to identify relevant context parameters from each dataset.
2) We expected that as CaCF methods were aware of the context (situation) the user was in, they could identify users whose movements are more relevant and useful for deriving location/place recommendations for the current user, and therefore, recommendations that are more appropriate can be generated. These experiments confirm this expectation, and show that introducing contextual information into the CF process can help to improve the recommendation quality.
3) The above results might suggest that among all the CaCF methods the contextual post-filtering (CaCF_Post) method and the contextual modeling (CaCF_Mdl) method might be more suitable for deriving context-aware location recommendations from location histories. As CaCF_Post has one more parameter to calibrate, in practical use, one might use CaCF_Mdl for a quick and “acceptable” result. For a better result, CaCF_Post can be then employed by calibrating both parameters, i.e., the neighborhood size N, and the context similarity

threshold δ.

5.6 Summary and conclusions

This chapter studied how contextual information can be integrated into non-contextual CF methods for deriving context-aware location recommendations from a large number of users' location histories. Specifically, we proposed a methodology for identifying context parameters that are relevant to the recommendation tasks (Section 5.2). We then explored a context similarity measure to measure the similarity between any two contexts (situations) (Section 5.3). We further integrated this context similarity measure to the non-contextual method (as developed in Chapter 4), and developed three kinds of context-aware CF methods (CaCF): contextual pre-filtering, contextual post-filtering, and contextual modeling (Section 5.4). With these CaCF methods, personalized and context-aware location recommendations can be provided in LBS applications.

We evaluated the CaCF methods, against the non-contextual CF method (nonCaCF) as developed in Chapter 4. Two real-world location history datasets were used for the evaluation: GPS trajectories at Delft city center, and GPS trajectories at Vienna zoo. These datasets reflect different scales of application scenarios. The results of the evaluation on the two datasets are similar, and demonstrate that: 1) When including contextual information in CF, choosing a suitable set of relevant context parameters is very important and may greatly affect the recommendation performance; 2) The identification of a set of relevant context parameters can be achieved by analyzing how users' aggregated movements differ in different situations; 3) The contextual post-filtering method achieves the best results, followed by the contextual modeling method, and finally the contextual pre-filtering method; 4) More importantly, all the CaCF methods perform better than nonCaCF. From these experiments, the following conclusion can be drawn: Including contextual information into the CF process provides users with more appropriate recommendations.

We believe that context-awareness plays a key role in LBS, and we plan to apply the idea of context-aware collaborative filtering (CaCF) to other LBS services, such as content selection and presentation. We are also interested in implementing the proposed CaCF methods into an LBS system, and therefore evaluating the CaCF methods in the field.

6 Conclusions and Future Work

6.1 Research contributions and conclusions

When visiting a new city, tourists often need help to effectively identify personally interesting places/locations from a potentially overwhelming set of choices. The task is further complicated by the physical attributes of the environments, as it takes time for people to move between places, and personally interesting places may be scattered throughout the environment (Bohnert 2010). Recently, the increasing availability of GPS-enabled devices and the rapid development of social media have led to the accumulation of a large number of location histories, such as GPS trajectories and trajectories constructed from users' "self-reported" information on the Internet. These location histories may reflect people's travel experiences in the environment. Research has shown that experiences from past users (especially similar ones) in similar contexts can help the current user to efficiently solve their problems (Wexelblat 1999, Zheng *et al.* 2011), e.g., choosing where to visit next or which route to take.

Motivated by the above two aspects, this research explored a methodology of deriving personalized and context-aware location recommendations from location histories in LBS applications. More specifically, we investigated *how human location histories, such as GPS trajectories and trajectories constructed from people's "self-reported" information on social media, and collaborative filtering can be integrated in LBS to provide users with personalized and context-aware location recommendations*. In the following, we summarize our research contributions. Similar to Bohnert (2010), we report on the summary by addressing the Sub-Questions as defined in Chapter 1.

Sub-Question 1 (Chapter 3): *How can a user's interests in various locations/places and motion behavior of visiting such locations, which are required for CF, be modeled and extracted from his/her location histories?*

Existing CF approaches often represent a user profile as a set of ratings given by the user on different items, which is insufficient to model information extracted from users' location histories. Specifically, the rating-based approach cannot model users' behavior of visiting various locations/places, e.g., in which orders these locations are visited, and transit time between locations. Based on the concepts of stops and moves, Chapter 3 explored a model of contextual user profile, which can be used to represent users' interests in various locations (reflected by their visits to these locations and the duration of these visits) as well as their behavior of visiting such locations. In order to extract meaningful user profiles from raw location histories, especially raw GPS trajectories, a duration-threshold-free SMoT (DTF-SMoT) method and a stay-point-based SMoT (SP-SMoT) method, which were extended from the state-of-the-art method, were developed.

Evaluation with two real-world GPS datasets shows that DTF-SMoT, which requires fewer parameters as inputs, achieves results that are comparable to the state-of-the-art method, i.e., the original SMoT method proposed by Alvares *et al.* (2007). Therefore, DTF-SMoT is a viable replacement of SMoT for scenarios where duration thresholds for candidate stops are rather difficult to define. On the other hand, SP-SMoT achieves results that are comparable to the SMoT approach, however, it also discovers stops that are unknown *a priori*, but may be potentially interesting to the application. Therefore, SP-SMoT is more useful for scenarios with insufficient background geographic information, in which defining a complete list of candidate stops is difficult.

The proposed model of contextual user profiles and the DTF-SMoT and SP-SMoT methods provide a basis for deriving recommendations from location histories. This non-intrusive user modeling technique can be also employed for other innovative applications, such as understanding of moving objects' behavior, activity recognition, and location-based social networking.

Sub-Question 2 (Chapter 4): *How can other users' interests in various locations and motion behavior of visiting such locations, as extracted from their location histories, be utilized to provide the current user with personalized location recommendations?*

CF is a promising technique for deriving location recommendations from location histories. Existing CF methods often work with rating-based user profiles, and are not suitable for making location recommendations from location histories. Chapter 4 addressed this issue, and investigated how information extracted from other users' location histories (i.e., their interests in various locations, and motion behavior of visiting such locations) can be aggregated for providing the current user with personalized location recommendations. More specifically, a novel user similarity measure, which considers the sequence property of movement (i.e., the order in which locations are visited), location popularity, duration at locations and transit time between locations, was developed to identify users who are similar to the current user. "Opinions" (i.e., movements) of these similar users were then aggregated to generate location recommendations to the current user.

We evaluated the proposed method with some benchmarking methods, via three real-world location history datasets (two GPS datasets and one dataset constructed from Flickr photos). The evaluation demonstrates that considering other people's movements, sequence relationships of locations visited, location popularity, duration at locations, and transit time between locations contributes to the improvement of recommendation quality. Among them, considering other people's movements achieves the biggest improvement. These experiments also confirmed that aggregating past users' location histories ("travel experiences") helps to provide the current users with personalized location recommendations.

The user similarity measure developed in this research also enables many other interesting applications, such as identifying potential friends in location-based social networks, and understanding interactions among people.

Sub-Question 3 (Chapter 5): *How can context-awareness be introduced to improve location recommendation in LBS?*

Context-awareness is a key aspect of LBS. However, existing methods for deriving recommendations from location histories often only consider users' current location and preferences, and fail to consider additional contextual information, such as weather and companion (i.e., with whom), which may be also relevant for the recommendation tasks. Chapter 5 addressed this issue, and investigated how the non-contextual CF method, as developed in Chapter 4, can be further improved by integrating contextual information.

We adopt the context definition proposed by Dey (2001, p. 5): "Context is any information that can be used to characterize the situation of an entity. An entity is a person, place, or object that is considered relevant to the interaction between a user and an application". We also adopt a "representational" view of context as described in Dourish (2004). According to this view, contexts/situations that the system may be in can be characterized or differentiated by a set of observable attributes (i.e., context parameters or dimensions), such as "weather". We developed a methodology for identifying context parameters that are relevant to the recommendation tasks. We then explored a statistics-based approach for measuring similarity between different contexts (situations). Three approaches were then designed to integrate the context similarity measure into the CF process, i.e., contextual pre-filtering, contextual post-filtering and contextual modeling. With these contextual methods, personalized and context-aware location recommendations can be provided in LBS.

The proposed methods were evaluated against the non-contextual methods as developed in Chapter 4, via two real-world location history datasets. The evaluation demonstrates that: 1) When including contextual information into CF, choosing a suitable set of relevant context parameters is very important and may greatly affect the recommendation performance; 2) The identification of a set of relevant context parameters can be achieved by analyzing how users' aggregated movements differ in different situations; 3) The contextual post-filtering method achieves the best results, followed by the contextual modeling method, and finally the contextual pre-filtering method; 4) More importantly, context-aware CF methods perform better than the non-contextual CF method. From these experiments, the following conclusion can be drawn: Including contextual information into CF improves recommendation quality, as it can provide users with more appropriate recommendations matching their context.

It is worth noting that, all the methods developed above are based on the concepts of stops and moves, which provide a fundamental and common conceptual framework for semantically processing different kinds of location histories, either recorded by devices/sensors or constructed from the Internet. In this sense, our methods are not restricted to a specific kind of location history.

Research addressing the above Sub-Questions can be then integrated to answer the ***overall research question*** of this research: "*How can human location histories (e.g., GPS trajectories and trajectories* *constructed from people's "self-reported" information on social media), and collaborative filtering (CF) be integrated in LBS to provide users with personalized and context-aware location recommendations?*"

Summarizing the whole research, we have gained insights about deriving personalized and context-aware recommendations from location histories in LBS applications. Specifically, we understand how relevant information about users' movement can be modeled and extracted from raw location histories, and how the extracted information can be aggregated and enriched with contextual information such as weather and companion (with whom) to provide personalized and context-aware recommendations. The overall solution can help tourists to identify personally interesting places from a potentially overwhelming set of choices when visiting an unfamiliar environment, such as a new city or a new museum.

As our approaches do not require an explicit representation of domain knowledge, they are very suitable for LBS applications, which might often need to provide services in scenarios with little (or no) available domain knowledge. Additionally, our approaches employ a non-intrusive user modeling technique and do not require users to state their interests and preferences explicitly, which are very promising in LBS applications, as LBS users are often involved in many tasks and activities during using their mobile devices. Furthermore, our approaches can provide users with personalized and context-aware recommendations, which are very welcome in LBS, as context-awareness plays a key role in LBS applications. In conclusion, we believe that the methods and findings discussed in this dissertation *significantly advance* the field of context-aware adaptation and personalization in LBS.

It is important to note that while this research focused on location recommendations, the insights gained in this research can be easily transferred and extended to other domains, e.g., product recommendations in mobile shopping guides, artwork recommendations in museums, friend recommendations in location-based social networks, human behavior understanding, and activity recognition. The principle of CF can be also applied to personalized and context-aware content selection/visualization in LBS.

6.2 Future work

This research has addressed some important questions in the area of personalization and context-aware adaptation. In this section, we list and motivate several promising areas for future development and research.

Evaluation with human participants. In the current research, approaches of leave-one-out validation on historical datasets were employed to evaluate the proposed algorithms. This evaluation can be improved by using experiments with human participants. For example, we can ask each participant to comment on the recommendation results generated by the proposed algorithms, like Zheng *et al.* (2011). We expect that similar results can be obtained in these kinds of human experiments.

Sequential recommendations. Current recommendation systems seldom provide sequence recommendations, e.g., a sequence of locations/places to visit next. In many LBS applications, this function is very important. For example, it might not make sense to recommend a location in the south, and then recommend another one in the north, and finally another one in the south again. In addition to travel distance, sequence recommendations might also need to consider the diversity of items in the recommended set, and co-occurrence interaction effects of different items (Hansen and Golbeck 2009). The proposed method in this research can be extended to provide sequence recommendations, by considering different constraints and interaction effects among items. Furthermore, the proposed method can be also extended to generate itinerary recommendation, for example, "recommend a two-day trip for visiting Vienna" or "recommend some exhibits to visit for a two-hour stay at the Albertina Museum".

Developing more comprehensive CaCF. In this research, we developed a methodology to identify relevant context parameters, and assumed the independence between context parameters. Even though the results of the experiments were very promising, this approach can be further improved by considering the correlation/multicollinearity[31] among different context parameters. Furthermore, the methodology can be extended to identify user-specific context parameters, and item-specific context parameters. With these, more relevant recommendations might be provided.

As mentioned in Section 3.1.2, the current research assumed that the context of a visit does not

[31] http://en.wikipedia.org/wiki/Multicollinearity

change during a user's visit. In order to further improve the recommendation performance, the current CaCF methods should be expanded to deal with dynamic contextual information.

Implicit profiling. In this research, a user's visit to a place/location (represented as a stop) and duration at this location is used to approximate his/her implicit interest rating for the location. However, the approximation of users' interest rating for a location/place can be improved by considering more aspects, such as users' travel behavior and activities at this location. We expect that having a more comprehensive approximation of users' implicit ratings in various locations will further improve the recommendation quality.

Recommendation user interfaces and explanations. In this research, we did not consider how to communicate the recommendation list to the end users. However, the way of communicating and visualizing the recommendation results also affects the usability of recommendation systems. Maps and Augmented Reality (AR) can be viable interfaces for showing these results. However, techniques for differentiating the results according to their predicted values (utilities) are still a topic of research. On the other hand, it is also interesting and useful to investigate methods that are able to generate explanations on why these locations are recommended. Again, how these explanations are communicated to the users is also an interesting research topic.

Integration with other recommendation techniques. In the current research, we employed CF for deriving recommendations from location histories. CF is an effective recommendation technique requiring little domain knowledge, which is very promising in LBS applications. However, it also suffers from the "cold-start" problem, as it cannot make recommendations for users who have little or no information available in the system (new user problem), and cannot recommend items which are newly added to the system (new item problem). This "cold-start" problem can be addressed by providing general and random recommendation (i.e., non-personalized). However, it can also be addressed by integrating other recommendation techniques (as introduced in Section 2.2), such as content-based, knowledge-based recommendations. As shown in other recommendation domains, such as movie recommendations and music recommendations, hybrid systems might help to further improve the recommendation quality (Adomavicius and Tuzhilin 2005). Therefore, it might be interesting to investigate how hybrid recommendation techniques can be used to provide more appropriate recommendations in LBS.

List of References

Adomavicius, G., Sankaranarayanan, R., Sen, S., and Tuzhilin, A., 2005. Incorporating contextual information in recommender systems using a multidimensional approach. *ACM Transactions on Information Systems*, 23 (1), 103–145.

Adomavicius, G. and Tuzhilin, A., 2005. Toward the next generation of recommender systems: a survey of the state-of-the-art and possible extensions. *IEEE Transactions on Knowledge and Data Engineering*, 17 (6), 734 – 749.

Adomavicius, G. and Tuzhilin, A., 2011. Context-Aware Recommender Systems. *In*: F. Ricci, L. Rokach, B. Shapira, and P.B. Kantor, eds. *Recommender Systems Handbook*. Springer US, 217–253.

Alvares, L.O., Bogorny, V., Kuijpers, B., de Macedo, J.A.F., Moelans, B., and Vaisman, A., 2007. A model for enriching trajectories with semantic geographical information. *In*: *Proceedings of the 15th annual ACM international symposium on Advances in geographic information systems*. New York, NY, USA: ACM, 22:1–22:8.

Anand, S. and Mobasher, B., 2007. Contextual Recommendation. *In*: B. Berendt, A. Hotho, D. Mladenic, and G. Semeraro, eds. *From Web to Social Web: Discovering and Deploying User and Content Profiles*. Springer Berlin / Heidelberg, 142–160.

Andrienko, G., Andrienko, N., and Heurich, M., 2011. An event-based conceptual model for context-aware movement analysis. *International Journal of Geographical Information Science*, 25 (9), 1347–1370.

Andrienko, N. and Andrienko, G., 2007. Designing visual analytics methods for massive collections of movement data. *Cartographica*, 42 (2), 117–138.

Bader, R., Neufeld, E., Woerndl, W., and Prinz, V., 2011. Context-aware POI recommendations in an automotive scenario using multi-criteria decision making methods. *In*: *Proceedings of the 2011 Workshop on Context-awareness in Retrieval and Recommendation*. New York, NY, USA: ACM, 23–30.

Baltrunas, L., 2011. Context-Aware Collaborative Filtering Recommender Systems. Ph.D. thesis. University of Bozen-Bolzano, Bozen-Bolzano, Italy.

Baus, J., Cheverst, K., and Kray, C., 2005. A Survey of Map-based Mobile Guides. *In*: L. Meng, T. Reichenbacher, and A. Zipf, eds. *Map-based Mobile Services*. Berlin/Heidelberg: Springer-Verlag, 193–209.

Bergroth, L., Hakonen, H., and Raita, T., 2000. A Survey of Longest Common Subsequence Algorithms. *In*: *Proceedings of the Seventh International Symposium on String Processing Information Retrieval (SPIRE'00)*. Washington, DC, USA: IEEE Computer Society, 39–.

Bogorny, V., Kuijpers, B., and Alvares, L.O., 2009. ST-DMQL: A Semantic Trajectory Data Mining Query Language. *International Journal of Geographical Information Science*, 23 (10), 1245–1276.

Bohnert, F., 2010. Non-Intrusive User Modelling and Behaviour Prediction in Museums. Ph.D. thesis. Monash University, Clayton, Australia.

Burke, R., 2002. Hybrid Recommender Systems: Survey and Experiments. *User Modeling and User-Adapted Interaction*, 12 (4), 331–370.

Cena, F., Console, L., Gena, C., Goy, A., Levi, G., Modeo, S., and Torre, I., 2006. Integrating heterogeneous adaptation techniques to build a flexible and usable mobile tourist guide. *AI Commun.*, 19 (4), 369–384.

Chen, C.-Y., Chang, B.R., and Huang, P.-S., 2013. Multimedia augmented reality information system for museum guidance. *Personal and Ubiquitous Computing*, 1–8.

Chen, G. and Kotz, D., 2000. *A Survey of Context-Aware Mobile Computing Research*. Hanover, NH, USA: Dartmouth College.

Chen, L., Özsu, M.T., and Oria, V., 2005. Robust and fast similarity search for moving object trajectories. *In*: *Proceedings of the 2005 ACM SIGMOD international conference on Management of data*. New York, NY, USA: ACM, 491–502.

Cheng, H., Arefin, M.S., Chen, Z., and Morimoto, Y., 2013. Place Recommendation Based on Users Check-in History for Location-Based Services. *International Journal of Networking and Computing*, 3 (2), 228–243.

Cheverst, K., Davies, N., Mitchell, K., and Smith, P., 2000. Providing Tailored (Context-Aware) Information to City Visitors. *In*: P. Brusilovsky, O. Stock, and C. Strapparava, eds. *Adaptive Hypermedia and Adaptive Web-Based Systems*. Springer Berlin / Heidelberg, 73–85.

De Choudhury, M., Feldman, M., Amer-Yahia, S., Golbandi, N., Lempel, R., and Yu, C., 2010. Automatic construction of travel itineraries using social breadcrumbs. *In*: *Proceedings of the 21st ACM conference on Hypertext and hypermedia*. New York, NY, USA: ACM, 35–44.

Coutaz, J., Crowley, J.L., Dobson, S., and Garlan, D., 2005. Context is key. *Communications of the ACM*, 48, 49.

Dean, D., 1996. *Museum Exhibition: Theory and Practice*. London: Routledge.

Delgado, J. and Ishii, N., 1999. Memory-Based Weighted-Majority Prediction for Recommender Systems. *In*: *Proc. ACM SIGIR '99 Workshop Recommender Systems: Algorithms and Evaluation*.

Desrosiers, C. and Karypis, G., 2011. A Comprehensive Survey of Neighborhood-based Recommendation Methods. *In*: F. Ricci, L. Rokach, B. Shapira, and P.B. Kantor, eds. *Recommender Systems Handbook*. Springer US, 107–144.

Dey, A.K., 2001. Understanding and Using Context. *Personal and Ubiquitous Computing*, 5, 4–7.

Dourish, P., 2004. What we talk about when we talk about context. *Personal and Ubiquitous Computing*, 8 (1), 19–30.

Felfernig, A., Friedrich, G., Jannach, D., and Zanker, M., 2011. Developing Constraint-based Recommenders. *In*: F. Ricci, L. Rokach, B. Shapira, and P.B. Kantor, eds. *Recommender Systems Handbook*. Springer US, 187–215.

Field, A., 2013. *Discovering statistics using IBM SPSS statistics*. London: SAGE.

Filippini Fantoni, S., 2003. Personalization through IT in Museums. Does it Really Work? - The Case of the Marble Museum Website. *In*: *Proceed- ings of the Seventh International Cultural Heritage Informatics Meeting (ICHIM-03)*.

Fischer, G., 1993. Shared Knowledge in Cooperative Problem-Solving Systems - Integrating Adaptive and Adaptable Components. *In*: M. Schneider-Hufschmidt, T. Kuehme, and U. Malinowski, eds. *Adaptive User Interfaces: Principles and practice*. Amsterdam: Elsevier Science Publishers B.V.

Focardi, S. and Cagnacci, C., 2013. Animal Movement. *In*: C. Renso, S. Spaccapietra, and E. Zimányi, eds. *Mobility Data: Modeling, Management, and Understanding*. Cambridge University Press.

Froehlich, J., Chen, M., Smith, I., and Potter, F., 2006. Voting with Your Feet: An Investigative Study of the Relationship Between Place Visit Behavior and Preference. *In*: P. Dourish and A. Friday, eds. *UbiComp 2006: Ubiquitous Computing*. Springer Berlin / Heidelberg, 333–350.

Gartner, G., 2013. Location-Based Services and the Role of Modern Cartography. *KN Kartographische Nachrichten - Journal of Cartography and Geographic Information*, (Special issue), 169–174.

Giannotti, F., Nanni, M., Pinelli, F., and Pedreschi, D., 2007. Trajectory pattern mining. *In*: *Proceedings of the 13th ACM SIGKDD international conference on Knowledge discovery and data mining*. New York, NY, USA: ACM, 330–339.

Gonzalez, H., Han, J., Li, X., Myslinska, M., and Sondag, J.P., 2007. Adaptive fastest path computation on a road network: a traffic mining approach. *In*: *Proceedings of the 33rd international conference on Very large data bases*. VLDB Endowment, 794–805.

Hagen, K. ten, Modsching, M., and Kramer, R., 2005. A Location Aware Mobile Tourist Guide Selecting and Interpreting Sights and Services by Context Matching. *In*: *Mobile and Ubiquitous Systems, Annual International Conference on*. Los Alamitos, CA, USA: IEEE Computer Society, 293–304.

Han, J., Lee, J.-G., Gonzalez, H., and Li, X., 2008. Mining Massive RFID, Trajectory, and Traffic Data Sets. *In*: *KDD Tutorial*. New York, NY, USA: ACM.

Han, S.-J. and Cho, S.-B., 2006. Predicting user's movement with a combination of self-organizing map and markov model. *In*: *Proceedings of the 16th international conference on Artificial Neural Networks - Volume Part II*. Berlin, Heidelberg: Springer-Verlag, 884–893.

Hanani, U., Shapira, B., and Shoval, P., 2001. Information Filtering: Overview of Issues, Research and Systems. *User Modeling and User-Adapted Interaction*, 11 (3), 203–259.

Hansen, D.L. and Golbeck, J., 2009. Mixing it up: recommending collections of items. *In*: *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*. New York, NY, USA: ACM, 1217–1226.

Hayes, C. and Cunningham, P., 2004. Context boosting collaborative recommendations. *Knowledge-Based Systems*, 17 (2–4), 131–138.

Herlocker, J.L. and Konstan, J.A., 2001. Content-independent task-focused recommendation. *Internet Computing, IEEE*, 5 (6), 40 –47.

Herlocker, J.L., Konstan, J.A., Borchers, A., and Riedl, J., 1999. An algorithmic framework for performing collaborative filtering. *In*: *Proceedings of the 22nd annual international ACM SIGIR conference on Research and development in information retrieval*. New York, NY, USA: ACM, 230–237.

Herlocker, J.L., Konstan, J.A., Terveen, L.G., and Riedl, J.T., 2004. Evaluating collaborative filtering recommender systems. *ACM Trans. Inf. Syst.*, 22 (1), 5–53.

Hofmann, T., 2003. Collaborative filtering via gaussian probabilistic latent semantic analysis. *In*: *Proceedings of the 26th annual international ACM SIGIR conference on Research and development in informaion retrieval*. New York, NY, USA: ACM, 259–266.

Horozov, T., Narasimhan, N., and Vasudevan, V., 2006. Using location for personalized POI recommendations in mobile environments. *In*: *International Symposium on Applications and the Internet, 2006. SAINT 2006*. 6 pp. –129.

Huang, H. and Gartner, G., 2009. Using Activity Theory to Identify Relevant Context Parameters. *In*: G. Gartner and K. Rehrl, eds. *Location Based Services and TeleCartography II*. Springer Berlin Heidelberg, 35–45.

Huang, H. and Gartner, G., 2010. A Survey of Mobile Indoor Navigation Systems. *In*: G. Gartner, F. Ortag, W. Cartwright, G. Gartner, L. Meng, and M.P. Peterson, eds. *Cartography in Central and Eastern Europe*. Springer Berlin Heidelberg, 305–319.

Hung, J.C., Huang, C.-H., and Hsu, V., 2012. An Adaptive Recommendation System for Museum

Navigation. *In*: R. Huang, A.A. Ghorbani, G. Pasi, T. Yamaguchi, N.Y. Yen, and B. Jin, eds. *Active Media Technology*. Springer Berlin Heidelberg, 534–542.

Janssens, D., Nanni, M., and Rinzivillo, S., 2013. Car Traffic Monitoring. *In*: C. Renso, S. Spaccapietra, and E. Zimányi, eds. *Mobility Data: Modeling, Management, and Understanding*. Cambridge University Press.

Johnston, R.J., 1998. Exogenous Factors and Visitor Behavior A Regression Analysis of Exhibit Viewing Time. *Environment and Behavior*, 30 (3), 322–347.

Karatzoglou, A., Amatriain, X., Baltrunas, L., and Oliver, N., 2010. Multiverse recommendation: n-dimensional tensor factorization for context-aware collaborative filtering. *In*: *Proceedings of the fourth ACM conference on Recommender systems*. New York, NY, USA: ACM, 79–86.

Kenteris, M., Gavalas, D., and Economou, D., 2011. Mytilene E-guide: a multiplatform mobile application tourist guide exemplar. *Multimedia Tools and Applications*, 54 (2), 241–262.

Keßler, C., 2010. Context-Aware Semantics-Based Information Retrieval. Ph.D. thesis. University of Münster, Münster, Germany.

Keßler, C., Raubal, M., and Wosniok, C., 2009. Semantic Rules for Context-Aware Geographical Information Retrieval. *In*: P. Barnaghi, K. Moessner, M. Presser, and S. Meissner, eds. *Smart Sensing and Context*. Springer Berlin Heidelberg, 77–92.

Kim, J.-B. and Park, C., 2011. Development of Mobile AR Tour Application for the National Palace Museum of Korea. *In*: R. Shumaker, ed. *Virtual and Mixed Reality - New Trends*. Springer Berlin Heidelberg, 55–60.

Koren, Y., 2008. Factorization meets the neighborhood: a multifaceted collaborative filtering model. *In*: *Proceedings of the 14th ACM SIGKDD international conference on Knowledge discovery and data mining*. New York, NY, USA: ACM, 426–434.

Krüger, A., Baus, J., Heckmann, D., Kruppa, M., and Wasinger, R., 2007. The adaptive web. *In*: P. Brusilovsky, A. Kobsa, and W. Nejdl, eds. Berlin, Heidelberg: Springer-Verlag, 521–549.

Letchner, J., Krumm, J., and Horvitz, E., 2006. Trip router with individualized preferences (TRIP): incorporating personalization into route planning. *In*: *Proceedings of the 18th conference on Innovative applications of artificial intelligence - Volume 2*. AAAI Press, 1795–1800.

Li, L.-H., Lee, F.-M., Chen, Y.-C., and Cheng, C.-Y., 2009. A multi-stage collaborative filtering approach for mobile recommendation. *In*: *Proceedings of the 3rd International Conference on Ubiquitous Information Management and Communication*. New York, NY, USA: ACM, 88–97.

Li, Q., Zheng, Y., Xie, X., Chen, Y., Liu, W., and Ma, W.-Y., 2008. Mining user similarity based on location history. *In*: *Proceedings of the 16th ACM SIGSPATIAL international conference on Advances in geographic information systems*. New York, NY, USA: ACM, 34:1–34:10.

Liao, L., Patterson, D.J., Fox, D., and Kautz, H., 2005. Building personal maps from gps data. *In*: *In Proceedings of IJCAI Workshop on Modeling Others from Observation*.

Lops, P., Gemmis, M., and Semeraro, G., 2011. Content-based Recommender Systems: State of the Art and Trends. *In*: F. Ricci, L. Rokach, B. Shapira, and P.B. Kantor, eds. *Recommender Systems Handbook*. Springer US, 73–105.

Luo, J., Joshi, D., Yu, J., and Gallagher, A., 2011. Geotagging in multimedia and computer vision—a survey. *Multimedia Tools and Applications*, 51 (1), 187–211.

Marketos, G., Damiani, M.L., Pelekis, N., Theodoridis, Y., and Yan, Z., 2013. Trajectory Collection and Reconstruction. *In*: C. Renso, S. Spaccapietra, and E. Zimányi, eds. *Mobility Data: Modeling, Management, and Understanding*. Cambridge University Press.

Marlin, B., 2003. Modeling User Rating Pro_les For Collaborative Filtering. *In*: *Proceedings of*

Advances in Neural Information Processing Systems (NIPS 2003). Vancouver and Whistler, British Columbia, Canada.

McNee, S.M., Riedl, J., and Konstan, J.A., 2006. Being accurate is not enough: how accuracy metrics have hurt recommender systems. *In*: *CHI '06 extended abstracts on Human factors in computing systems*. New York, NY, USA: ACM, 1097–1101.

Nanni, M., 2013. Mobility data mining. *In*: C. Renso, S. Spaccapietra, and E. Zimányi, eds. *Mobility Data: Modeling, Management, and Understanding*. Cambridge University Press.

Nichols, D.M., 1997. Implicit Rating and Filtering. *In*: *Proceedings of the 5th DELOS Workshop on Filtering and Collaborative Filtering*. Budapest, Hungary, 31–36.

Nivala, A.-M. and Sarjakoski, L.T., 2003. An Approach to Intelligent Maps: Context Awareness. Presented at the Workshop 'HCI in mobile Guides', Udine, Italy.

Oku, K., Nakajima, S., Miyazaki, J., and Uemura, S., 2006. Context-Aware SVM for Context-Dependent Information Recommendation. *In*: *Mobile Data Management 2006*. 109.

Oppermann, R. and Specht, M., 1999. A Nomadic Information System for Adaptive Exhibition Guidance. *In*: *Proc. of the International Conference on Hypermedia and Interactivity in Museums*. 103–109.

Palma, A.T., Bogorny, V., Kuijpers, B., and Alvares, L.O., 2008. A clustering-based approach for discovering interesting places in trajectories. *In*: *Proceedings of the 2008 ACM symposium on Applied computing*. New York, NY, USA: ACM, 863–868.

Panniello, U. and Gorgoglione, M., 2012. Incorporating context into recommender systems: an empirical comparison of context-based approaches. *Electronic Commerce Research*, 12 (1), 1–30.

Panniello, U., Tuzhilin, A., Gorgoglione, M., Palmisano, C., and Pedone, A., 2009. Experimental comparison of pre- vs. post-filtering approaches in context-aware recommender systems. *In*: *Proceedings of the third ACM conference on Recommender systems*. New York, NY, USA: ACM, 265–268.

Park, M.-H., Hong, J.-H., and Cho, S.-B., 2007. Location-Based Recommendation System Using Bayesian User's Preference Model in Mobile Devices. *In*: J. Indulska, J. Ma, L. Yang, T. Ungerer, and J. Cao, eds. *Ubiquitous Intelligence and Computing*. Springer Berlin / Heidelberg, 1130–1139.

Pavlov, D. and Pennock, D.M., 2002. A Maximum Entropy Approach to Collaborative Filtering in Dynamic, Sparse, High-Dimensional Domains. *In*: *Proceeding of Advances in Neural Information Processing Systems*. Vancouver, British Columbia, Canada.

Pazzani, M. and Billsus, D., 2007. Content-Based Recommendation Systems. *In*: P. Brusilovsky, A. Kobsa, and W. Nejdl, eds. *The Adaptive Web*. Springer Berlin / Heidelberg, 325–341.

Pombinho, P., Carmo, M.B., and Afonso, A.P., 2012. Context aware point of interest adaptive recommendation. *In*: *Proceedings of the 2nd Workshop on Context-awareness in Retrieval and Recommendation*. New York, NY, USA: ACM, 30–33.

Raper, J., Gartner, G., Karimi, H., and Rizos, C., 2007a. A critical evaluation of location based services and their potential. *Journal of Location Based Services*, 1 (1), 5–45.

Raper, J., Gartner, G., Karimi, H., and Rizos, C., 2007b. Applications of location–based services: a selected review. *Journal of Location Based Services*, 1 (2), 89–111.

Raubal, M. and Panov, I., 2009. A Formal Model for Mobile Map Adaptation. *In*: G. Gartner and K. Rehrl, eds. *Location Based Services and TeleCartography II*. Springer Berlin Heidelberg, 11–34.

Reichenbacher, T., 2004. Mobile cartography - adaptive visualisation of geographic information on

mobile devices. Ph.D. thesis. Technical University of Munich.
Renso, C., Baglioni, M., Macedo, J.A.F. de, Trasarti, R., and Wachowicz, M., 2013. How you move reveals who you are: understanding human behavior by analyzing trajectory data. *Knowledge and Information Systems*, 1–32.
Renso, C. and Trasarti, R., 2013. Understaning human mobility using Mobility Data Mining. *In*: C. Renso, S. Spaccapietra, and E. Zimányi, eds. *Mobility Data: Modeling, Management, and Understanding*. Cambridge University Press.
Resnick, P., Iacovou, N., Suchak, M., Bergstrom, P., and Riedl, J., 1994. GroupLens: an open architecture for collaborative filtering of netnews. *In*: *Proceedings of the 1994 ACM conference on Computer supported cooperative work*. New York, NY, USA: ACM, 175–186.
Resnick, P. and Varian, H.R., 1997. Recommender systems. *Commun. ACM*, 40 (3), 56–58.
Retscher, G., 2007. Augmentation of Indoor Positioning Systems with a Barometric Pressure Sensor for Direct Altitude Determination in a Multi-storey Building. *Cartography and Geographic Information Science*, 34 (4), 305–310.
Ricci, F., Rokach, L., and Shapira, B., 2011. Introduction to Recommender Systems Handbook. *In*: F. Ricci, L. Rokach, B. Shapira, and P.B. Kantor, eds. *Recommender Systems Handbook*. Springer US, 1–35.
Rinzivillo, S., Siqueira, F. de L., Gabrielli, L., Renso, C., and Bogorny, V., 2013. Where Have You Been Today? Annotating Trajectories with DayTag. *In*: M.A. Nascimento, T. Sellis, R. Cheng, J. Sander, Y. Zheng, H.-P. Kriegel, M. Renz, and C. Sengstock, eds. *Advances in Spatial and Temporal Databases*. Springer Berlin Heidelberg, 467–471.
Rocha, J.A.M.R., Oliveira, G., Alvares, L.O., Bogorny, V., and Times, V.C., 2010. DB-SMoT: A direction-based spatio-temporal clustering method. *In*: *Intelligent Systems (IS), 2010 5th IEEE International Conference*. 114 –119.
Salton, G. and McGill, M.J., 1986. *Introduction to Modern Information Retrieval*. New York, NY, USA: McGraw-Hill, Inc.
Sarjakoski, T. and Sarjakoski, L.T., 2005. *The GiMoDig public final report*.
Sarwar, B., Karypis, G., Konstan, J., and Riedl, J., 2000. Analysis of recommendation algorithms for e-commerce. *In*: *Proceedings of the 2nd ACM conference on Electronic commerce*. New York, NY, USA: ACM, 158–167.
Schafer, J.B., Konstan, J., and Riedi, J., 1999. Recommender systems in e-commerce. *In*: *Proceedings of the 1st ACM conference on Electronic commerce*. New York, NY, USA: ACM, 158–166.
Schilit, B., Adams, N., and Want, R., 1994. Context-Aware Computing Applications. *In*: *Proceedings of the 1994 First Workshop on Mobile Computing Systems and Applications*. Washington, DC, USA: IEEE Computer Society, 85–90.
Schmidt, A., 2002. Ubiquitous Computing - Computing in Context. PhD Thesis. Lancaster University, UK.
Schmidt, A., Beigl, M., and Gellersen, H.-W., 1999. There is more to context than location. *Computers & Graphics*, 23 (6), 893–901.
Schmidt-Belz, B., Nick, A., Poslad, S., and Zipf, A., 2002. Personalized and Location-based Mobile Tourism Services.
Van Setten, M., Pokraev, S., and Koolwaaij, J., 2004. Context-Aware Recommendations in the Mobile Tourist Application COMPASS. *In*: P. De Bra and W. Nejdl, eds. *Adaptive Hypermedia and Adaptive Web-Based Systems*. Springer Berlin / Heidelberg, 515–548.
Singhal, A., 2001. Modern information retrieval: a brief overview. *Bulletin of the IEEE Computer*

Society Technical Committee on Data Engineering, 24 (4), 35–42.

Spaccapietra, S., Parent, C., Damiani, M.L., de Macedo, J.A., Porto, F., and Vangenot, C., 2008. A conceptual view on trajectories. *Data Knowl. Eng.*, 65 (1), 126–146.

Spaccapietra, S., Parent, C., and Spinsanti, L., 2013. Trajectories and their representations. *In*: C. Renso, S. Spaccapietra, and E. Zimányi, eds. *Mobility Data: Modeling, Management, and Understanding*. Cambridge University Press.

Sparacino, F., 2002. Real-time sensor-driven understanding of visitors' interests for personalized visually-augmented museum experiences. *In*: *Proceedings of the 11th International Conference on Museums and the Web (MW-02)*.

Van der Spek, S., van Schaick, J., de Bois, P., and de Haan, R., 2009. Sensing Human Activity: GPS Tracking. *Sensors*, 9 (4), 3033–3055.

De Spindler, A., Norrie, M.C., Grossniklaus, M., and Signer, B., 2006. Spatio-Temporal Proximity as a Basis for Collaborative Filtering in Mobile Environments. *In*: *Proceedings of the Workshop on Ubiquitous Mobile Information and Collaboration Systems (UMICS 2006) at the 18th International Conference on Advanced Information Systems Engineering (CAiSE 2006)*. Luxembourg, Grand Duchy of Luxembourg.

Stock, O. and Zancanaro, M., 2007. *PEACH - Intelligent Interfaces for Museum Visits*. Springer.

Takács, G., Pilászy, I., Németh, B., and Tikk, D., 2007. Major components of the gravity recommendation system. *SIGKDD Explor. Newsl.*, 9 (2), 80–83.

Takeuchi, Y. and Sugimoto, M., 2006. CityVoyager: An Outdoor Recommendation System Based on User Location History. *In*: J. Ma, H. Jin, L. Yang, and J. Tsai, eds. *Ubiquitous Intelligence and Computing*. Springer Berlin / Heidelberg, 625–636.

Tallon, L. and Walker, K., 2008. *Digital technologies and the museum experience: handheld guides and other media*. Lanham: AltaMira Press.

Versichele, M., Neutens, T., Delafontaine, M., and Van de Weghe, N., 2012. The use of Bluetooth for analysing spatiotemporal dynamics of human movement at mass events: A case study of the Ghent Festivities. *Applied Geography*, 32 (2), 208–220.

Versichele, M., Neutens, T., and Van de Weghe, N., 2013. Person monitoring with Bluetooth tracking. *In*: C. Renso, S. Spaccapietra, and E. Zimányi, eds. *Mobility Data: Modeling, Management, and Understanding*. Cambridge University Press.

Wakkary, R. and Evernden, D., 2005. Museum As Ecology: A Case Study Analysis Of An Ambient Intelligent Museum Guide. *In*: *Proceedings of the 14th International Conference on Museums and the Web (MW- 05)*.

Weakliam, J., Wilson, D., and Bertolotto, M., 2008. Personalising Map Feature Content for Mobile Map Users. *In*: L. Meng, A. Zipf, and S. Winter, eds. *Map-based Mobile Services*. Springer Berlin Heidelberg, 125–145.

Wexelblat, A., 1999. Footprints: Interaction history for digital objects. Ph.D. thesis. MIT Program in Media Arts & Sciences.

Wiesenhofer, S., Feiertag, H., Ray, M., Paletta, L., Luley, P., Alexander, A., Mathias, S., Josef, R., and Beyer, P., 2007. Mobile City Explorer: An innovative GPS and Camera Phone Based Travel Assistant for City Tourists. *In*: P.D.G. Gartner, P.D.W. Cartwright, and P.D.M.P. Peterson, eds. *Location Based Services and TeleCartography*. Springer Berlin Heidelberg, 557–573.

Winograd, T., 2001. Architectures for context. *Hum.-Comput. Interact.*, 16 (2), 401–419.

Xiao, X., Zheng, Y., Luo, Q., and Xie, X., 2012. Inferring Social Ties between Users with Human Location History. *Journal of Ambient Intelligence and Humanized Computing*.

Yan, P. and Zeng, D.D., 2009. Spatial Movement Pattern Discovery with LCS Based Path Similarity Measure. *In*: *Proceedings of the 15th Americas Conference on Information Systems*. Presented at the AMCIS 2009, San Francisco, California, USA.

Yan, Z., Chakraborty, D., Parent, C., Spaccapietra, S., and Aberer, K., 2011. SeMiTri: a framework for semantic annotation of heterogeneous trajectories. *In*: *Proceedings of the 14th International Conference on Extending Database Technology*. New York, NY, USA: ACM, 259–270.

Yang, W.-S. and Hwang, S.-Y., 2013. iTravel: A recommender system in mobile peer-to-peer environment. *Journal of Systems and Software*, 86 (1), 12–20.

Yap, G.-E., Tan, A.-H., and Pang, H.-H., 2005. Dynamically-optimized context in recommender systems. *In*: *Proceedings of the 6th international conference on Mobile data management*. New York, NY, USA: ACM, 265–272.

Yap, G.-E., Tan, A.-H., and Pang, H.-H., 2007. Discovering and Exploiting Causal Dependencies for Robust Mobile Context-Aware Recommenders. *IEEE Transactions on Knowledge and Data Engineering*, 19 (7), 977–992.

Ye, J., 2009. Exploiting Semantics with Situation Lattices in Pervasive Computing. PhD Thesis. University College Dublin.

Ye, Y., Zheng, Y., Chen, Y., Feng, J., and Xie, X., 2009. Mining Individual Life Pattern Based on Location History. *In*: *Tenth International Conference on Mobile Data Management: Systems, Services and Middleware, 2009. MDM '09*. 1 –10.

Yılmaz, Ö. and Erdur, R.C., 2012. iConAwa – An intelligent context-aware system. *Expert Systems with Applications*, 39 (3), 2907–2918.

Yoon, H., Zheng, Y., Xie, X., and Woo, W., 2012. Social itinerary recommendation from user-generated digital trails. *Personal Ubiquitous Comput.*, 16 (5), 469–484.

Yu, C.-C. and Chang, H., 2013. Towards Context-Aware Recommendation for Personalized Mobile Travel Planning. *In*: P.C. Vinh, N.M. Hung, N.T. Tung, and J. Suzuki, eds. *Context-Aware Systems and Applications*. Springer Berlin Heidelberg, 121–130.

Yuan, J., Zheng, Y., Zhang, C., Xie, W., Xie, X., Sun, G., and Huang, Y., 2010. T-drive: driving directions based on taxi trajectories. *In*: *Proceedings of the 18th SIGSPATIAL International Conference on Advances in Geographic Information Systems*. New York, NY, USA: ACM, 99–108.

Yuan, Y. and Raubal, M., 2012. Extracting Dynamic Urban Mobility Patterns from Mobile Phone Data. *In*: N. Xiao, M.-P. Kwan, M.F. Goodchild, and S. Shekhar, eds. *Geographic Information Science*. Springer Berlin Heidelberg, 354–367.

Zheng, V.W., Zheng, Y., Xie, X., and Yang, Q., 2012. Towards mobile intelligence: Learning from GPS history data for collaborative recommendation. *Artificial Intelligence*, 184–185, 17–37.

Zheng, Y., Chen, Y., Li, Q., Xie, X., and Ma, W.-Y., 2010. Understanding transportation modes based on GPS data for web applications. *ACM Trans. Web*, 4 (1), 1:1–1:36.

Zheng, Y., Xie, X., and Ma, W.-Y., 2010. GeoLife: A Collaborative Social Networking Service among User, Location and Trajectory. *IEEE Data Eng. Bull.*, 33 (2), 32–39.

Zheng, Y., Zhang, L., Ma, Z., Xie, X., and Ma, W.-Y., 2011. Recommending friends and locations based on individual location history. *ACM Trans. Web*, 5 (1), 5:1–5:44.

Zheng, Y., Zhang, L., Xie, X., and Ma, W.-Y., 2009. Mining interesting locations and travel sequences from GPS trajectories. *In*: *Proceedings of the 18th international conference on World wide web*. New York, NY, USA: ACM, 791–800.

Ziebart, B.D., Maas, A.L., Dey, A.K., and Bagnell, J.A., 2008. Navigate like a cabbie: probabilistic reasoning from observed context-aware behavior. *In*: *Proceedings of the 10th international* *conference on Ubiquitous computing*. New York, NY, USA: ACM, 322–331.

Curriculum Vitae

23.03.1983	Born in Puning, Guangdong, China
2000-2004	Bachelor studies at the Department of Computer Science, South China Normal University, Guangzhou, China
2004-2006	Master of Science (MSc) studies on Computer Software and Theory, Department of Computer Science, South China Normal University
2007-2013	Doctoral studies on Environmental Science, Guangzhou Institute of Geochemistry, Chinese Academy of Sciences, Guangzhou, China Research topic: Web-based Spatial Environmental Information Representation and Analysis (Defensed on Jan. 2013)
02.2010-07.2010	Recipient of the Marshall Plan Scholarship, and visiting scholar of the University of California, Santa Barbara
2007-	Research assistant at Research Group Cartography, Vienna University of Technology, Vienna, Austria
2008-	Doctoral studies at Research Group Cartography of Department of Geodesy and Geoinformation, Vienna University of Technology, Austria