TALLINN UNIVERSITY OF TECHNOLOGY
DOCTORAL THESIS
6/2020

Place Recommendation with Geo-tagged Photos

PRIIT JÄRV

PRESS

TALLINN UNIVERSITY OF TECHNOLOGY
School of Information Technologies
Department of Software Science
The dissertation was accepted for the defense of the degree of Doctor of Philosophy in Computer Science on 10 February 2020

Supervisor: Professor Tanel Tammet,
Department of Software Science
Tallinn University of Technology
Tallinn, Estonia

Opponents: Professor Francesco Ricci,
Free University of Bozen-Bolzano,
Bolzano, Italy

Professor Christian S. Jensen,
University of Aalborg,
Aalborg, Denmark

Defense of the thesis: 24 March 2020, Tallinn

Declaration:
Hereby I declare that this doctoral thesis, my original investigation and achievement, submitted for the doctoral degree at Tallinn University of Technology, has not been submitted for any academic degree elsewhere.

Priit Järv ______________________________
signature

European Union
European Regional
Development Fund

Investing
in your future

ISSN 2585-6898 (publication)
ISBN 978-9949-83-533-1 (publication)
ISSN 2585-6901 (PDF)
ISBN 978-9949-83-534-8 (PDF)

TALLINNA TEHNIKAÜLIKOOL
DOKTORITÖÖ
6/2020

Geomärgistega fotode kaevandamine reisisoovitusteks

PRIIT JÄRV

TALTECH KIRJASTUS

Contents

List of Publications

The work in this thesis is based on the following publications.

Publication I: T. Tammet, A. Luberg, and P. Järv. Sightsmap: Crowd-sourced popularity of the world places. In L. Cantoni and Z. P. Xiang, editors, *Information and Communication Technologies in Tourism 2013*, pages 314–325. Springer Berlin Heidelberg, 2013

Publication II: P. Järv. Extracting human mobility data from geo-tagged photos. In *Proceedings of the PredictGIS'17:1st ACM SIGSPATIAL Workshop on Prediction of Human Mobility*, pages 4:1–4:7, 2017

Publication III: P. Järv, T. Tammet, and M. Tall. Hierarchical regions of interest. In *19th IEEE International Conference on Mobile Data Management, MDM 2018, Aalborg, Denmark, June 25-28, 2018*, pages 86–95. IEEE Computer Society, 2018

Publication IV: P. Järv. Predictability limits in session-based next item recommendation. In T. Bogers, A. Said, P. Brusilovsky, and D. Tikk, editors, *Proceedings of the 13th ACM Conference on Recommender Systems, RecSys 2019, Copenhagen, Denmark, September 16-20, 2019.*, pages 146–150. ACM, 2019

Author's Contributions to the Publications

- **I** In Publication I, I created the recommender system to recommend trip itineraries on the sightsmap.com pages. This paper is included to establish the concept of mining geo-tagged photos for places of interest and visitor behavior.
- **II** In Publication II, I was the main author of the paper.
- **III** In Publication III, I was the main author, collected the data, designed and ran the experiments and analyzed the results. The co-authors contributed to the literature review, preliminary experiments and conducting the surveys.
- **IV** In Publication IV, I was the main author of the paper.

Other Publications

Other publications that the author has contributed to during the studies at the Tallinn University of Technology.

Publication V: A. Luberg, T. Tammet, and P. Järv. Smart city: A rule-based tourist recommendation system. In R. Law, M. Fuchs, and F. Ricci, editors, *Information and Communication Technologies in Tourism 2011 - Proceedings of the International Conference in Innsbruck, Austria, January 26-28, 2011*, pages 51–62. Springer Vienna, 2011

Publication VI: A. Luberg, T. Tammet, and P. Järv. Extended triple store structure used in recommender system. In F. Morvan, A. M. Tjoa, and R. Wagner, editors, *2011 Database and Expert Systems Applications, DEXA, International Workshops, Toulouse, France, August 29 - Sept. 2, 2011*, pages 539–543. IEEE Computer Society, 2011

Publication VII: A. Luberg, P. Järv, K. Schoefegger, and T. Tammet. Context-aware and multilingual information extraction for a tourist recommender system. In S. N. Lindstaedt and M. Granitzer, editors, *I-KNOW 2011, 11th International Conference on Knowledge Management and Knowledge Technologies, Graz, Austria, September 7-9, 2011*, page 13. ACM, 2011

Publication VIII: A. Luberg, P. Järv, and T. Tammet. Information extraction for a tourist recommender system. In M. Fuchs, F. Ricci, and L. Cantoni, editors, *Information and Communication Technologies in Tourism 2012, ENTER 2012, Proceedings of the International Conference in Helsingborg, Sweden, January 25-27, 2012.*, pages 332–343. Springer, 2012

Publication IX: A. Luberg, M. Granitzer, H. Wu, P. Järv, and T. Tammet. Information retrieval and deduplication for tourism recommender sightsplanner. In D. D. Burdescu, R. Akerkar, and C. Badica, editors, *2nd International Conference on Web Intelligence, Mining and Semantics, WIMS '12, Craiova, Romania, June 6-8, 2012*, pages 50:1–50:11. ACM, 2012

Publication X: K. Tomingas, T. Tammet, M. Kliimask, and P. Järv. Automating component dependency analysis for enterprise business intelligence. In M. D. Myers and D. W. Straub, editors, *Proceedings of the International Conference on Information Systems - Building a Better World through Information Systems, ICIS 2014, Auckland, New Zealand, December 14-17, 2014*. Association for Information Systems, 2014

Publication XI: K. Tomingas, P. Järv, and T. Tammet. Discovering data lineage from data warehouse procedures. In A. L. N. Fred, J. L. G. Dietz, D. Aveiro, K. Liu, J. Bernardino, and J. Filipe, editors, *Proceedings of the 8th International Joint Conference on Knowledge Discovery, Knowledge Engineering and Knowledge Management (IC3K 2016) - Volume 1: KDIR, Porto - Portugal, November 9 - 11, 2016.*, pages 101–110. SciTePress, 2016

Publication XII: K. Tomingas, P. Järv, and T. Tammet. Computing data lineage and business semantics for data warehouse. In A. L. N. Fred, J. L. G. Dietz, D. Aveiro, K. Liu, J. Bernardino, and J. Filipe, editors, *Knowledge Discovery, Knowledge Engineering and Knowledge Management - 8th International Joint Conference, IC3K 2016, Porto, Portugal, November 9-11, 2016, Revised Selected Papers*, volume 914 of *Communications in Computer and Information Science*, pages 101–124. Springer, 2016

Introduction

Recommender systems are software applications designed to connect people with objects and activities they are interested in. The overabundance of information, products and services available can make it difficult and time-consuming to find goods, travel destinations or entertainment. Recommender systems determine the interests of the user and deliver a shortlist of recommendations.

While their purpose is to improve user experience, recommender systems also serve business interests. The system anticipates what users may wish to purchase or view and increases the likelihood of purchases or other use of services by displaying such content. For this reason, recommendations are often given without the user specifically requesting them.

The traditional platforms for recommender systems are online retail and entertainment websites. Amazon.com adopted recommendation technology early on. Features such as "users who bought this item also bought ..." made the general public aware of recommender systems and rapidly became common on other e-commerce sites. Amazon's success, however, can be attributed to integrating recommendations throughout the shopping experience and scaling the recommendation technology to its large customer base [71]. Netflix used personalized recommendations in online DVD rental. Both their and Amazon's recommendations relied on relatively simple models to predict the ratings of items based on the ratings that similar users had given. Netflix attracted significant attention to recommender systems by offering a $1000000 prize for improving their recommendation algorithms [41].

Due to the early successes and publicity of recommender systems, it was expected that they would become ubiquitous wherever users faced making choices from a large set of options. The adoption of recommender systems, however, has not spread as quickly as initially expected. This is because producing recommendations that users perceive as experience-enhancing can be difficult depending on the application.

This thesis describes approaches for recommending tourist destinations and trip itineraries. Research on complete tourist recommender systems was published as early as 2004 and included favorable evaluations [121]. Machine-generated recommendations for tourists, however, have not been as widely adopted as social recommendations. People still prefer the advice of other people when going on a trip. In case their own social group has no previous experience with the intended destination, travel advice can be looked up online, for example using TripAdvisor[1] or Foursquare[2]. These applications allow users to browse and discover travel destinations and see how other users rate them. The widespread use of TripAdvisor ratings on websites that feature travel content provides evidence that users perceive peer recommendations as trustworthy.

Automated recommender systems have some advantages as they can provide features such as context awareness and planning automation. However, the issue of building user trust still needs to be tackled. One requirement for this is that the recommender output must be accurate. This means that users must consider the recommendations plausible and the result of accepting a recommendation must be satisfying.

The first challenge in developing tourist recommenders is that, unlike recommendation scenarios for retail or video content, the recommendation is not a single item. The trip itinerary is a list of multiple places that must fit together as a sequence. Recommending sequences is an active direction of ongoing research.

[1] `http://tripadvisor.com`

[2] `http://foursquare.com`

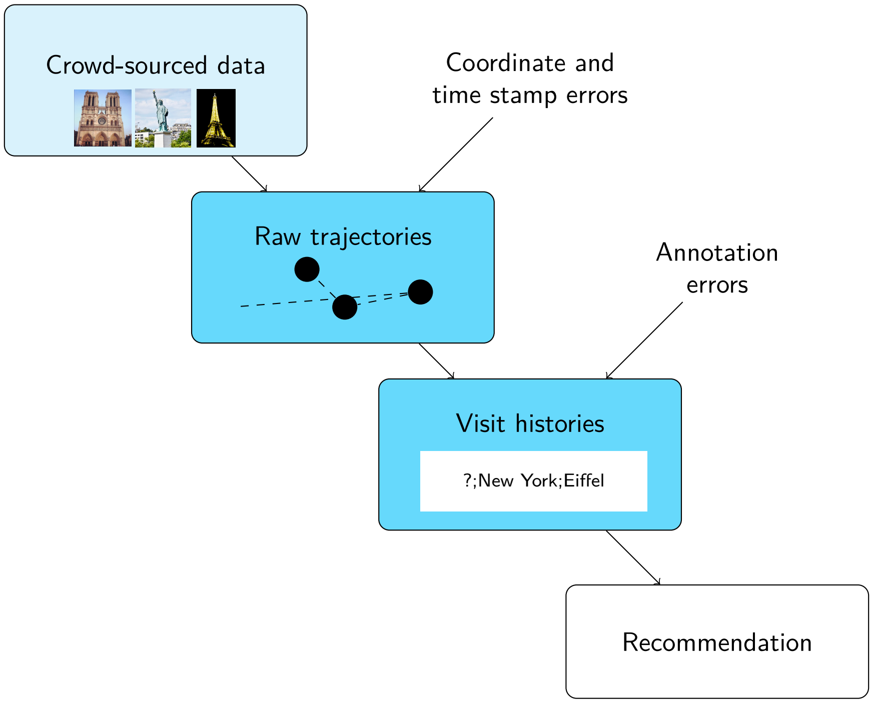

Figure 1: Workflow from crowd-sourced data to recommendation. At each stage, errors can accumulate and affect the quality of the recommendation.

To build successful behavioral models for recommendation, additional challenges include integrating heterogeneous data sources, context awareness and personalization. A recommender system must solve the problem of obtaining knowledge on places to recommend as well as how those places relate to different users and contexts.

This thesis focuses on improving the accuracy of recommender systems by improving the quality of the data used to build recommendation models. The availability of geo-tagged, time stamped and annotated information on social networks makes it possible to extract traces of individual movements and learn the behavior of tourists. Geo-tagged photos are available on large scale, enabling us to target many geographic areas and obtain enough information to learn behavioral patterns. In the thesis, we develop and evaluate methods that ensure the accuracy of the extracted behavioral patterns.

Figure 1 describes the typical recommendation flow using geo-tagged photos. Because the photos have associated time stamps and coordinates, it is possible to trace the movement of the users that posted the photos. This allows extraction of movement trajectories, which can be used to infer the places that the users visited. The visiting patterns obtained this way are used to build behavioral models that can predict a suitable place to visit, given the history of previous visits of the user.

At each stage, there are potential errors that reduce the quality of the final recommendation. The information shared by users is noisy. Due to a multitude of factors, photos may have invalid coordinates or time stamps. The data is sparse and not annotated. To transform the sequences of photos to place visiting histories, the photos have to be annotated with place names. In the thesis, we develop noise filtering heuristics and new methods for accurate semantic annotation of tourist trajectories to transform them into sequences of place visits.

While many proposals of complete recommender solutions using geo-tagged photos have been published, the existing literature has focused on the performance of the predictive models and there has been no systematic evaluation of the accuracy and impact of the methods used in the data preparation stage. Our contribution includes this evaluation that shows the impact of data quality.

We evaluate the techniques we propose both by their performance at their specific task, as well as by measuring their effect on the outcome of recommendation. Throughout the thesis, we assume that the task of place recommendation is sequential. The recommended places have to form or fit in a coherent sequence. Additionally, we assume that not much information is available or required from the users themselves, which allows recommendations without specific queries made by the user or without requiring the user to identify themselves.

Predictive models for recommendation have received a lot of attention in the research community, so we apply existing well known models and compare their output when given training data prepared using different methods. Assuming that the quality of the training data affects the predictive performance of recommenders, we additionally introduce a model-agnostic predictability metric for datasets.

Recommender System Concepts

In the literature about recommender systems it is conventional to use the following terminology. Throughout the thesis we will use these conventional meanings of recommendation concepts.

- *Users* are the consumers of recommender systems. They interact with the system to receive recommendations. At the same time, the system collects information about the users and includes it in the recommendation *model*. The information collected usually includes either explicit or implicit preferences of the user. Explicit preferences are choices or selections about the content the user wishes to receive. Implicit preferences are derived from the behavioral or interaction history of the user.
- *Items* are what is being recommended and can include goods for sale, media for entertainment or places to visit.
- The recommendation *model* usually includes both users and items and describes the relations between them, such as what items the user has liked earlier and what other users the user interacts with or is similar to. Advanced models may contain contextual information about the location, time, weather, events and more. The task of the model is to predict which items the user would be most interested in.
- *Collaborative filtering* (CF) is the technique where the behavior of similar users is used in recommending items. An intuitive model for this is a user-item graph, where edges connect users and items. Similar users can then be found by looking for users that have in the past preferred similar items. For example, if user A has liked the movie "Titanic" then users B and C who have liked "Titanic" are considered similar. If B and C both also watched "The Beach", that A has not watched yet, this movie could be recommended to user A. Collaborative filtering may be enhanced with including content and contextual information, for example the genre and the lead actors of a movie or the weather, time and season in place recommendation.

Trip and Next Place Recommendation

We define trip recommendation as the task of recommending a sequence of places in an area, such as a city or geographic region. Real-world applications need to deal with additional aspects, such as travel and accommodation logistics, monetary and time budgets and group recommendation. In the thesis we focus on core challenges that any recommender system would need to solve: data acquisition and knowledge engineering. High quality knowledge about the places and the interests of the users are the basis of building a recommendation model.

The same core challenges need to be solved in next place recommendation - the task of recommending the next place to visit based on the user's recent history. Some research has also focused on recommending locations in general, usually in the context of social networks like Foursquare.

We also consider the scenario where the user is not well known to the system and the recommendation has to be done based on the information the user has implicitly given by their current interaction with the recommender system. Recently, more attention in recommender systems research has focused on this problem as it is important to reach users who may consider it inconvenient to provide their full interests or simply wish to interact with the system anonymously.

Finally, we consider trip and next place recommendation to be a sequence-aware task: both the user's visiting history and the way recommended places fit together, should be taken into account.

We use the following common terminology for concepts in trip and next place recommendation.

- *Place* is the item to be recommended: a shop, landmark, museum, ride, activity and similar small-scale objects. In the literature these are also frequently called locations or points of interests. Our approach also involves larger-scale objects such as parks, lakes and neighborhoods. We use the term *area* to refer to such larger-scale features when it is necessary to differentiate them from small-scale objects.

- *Point of interest* (POI) is a place defined by its geographical coordinates, name and additional metadata, such as place category or opening times. Points of interests are ubiquitously used with location related applications and databases of POIs are commonly used to recommend nearby places or identify the location of the user.

- *Region of interest* (ROI) is defined by its spatial extent, such as a polygon on the map, combined with metadata similar to points of interest. In some of the literature, the term area of interest (AoI) or zone of interest (ZOI) is used. We use ROIs similarly to POIs as they capture the geographical features of places and areas more accurately.

- *Stay point* is a point or a small area where the movement along the trajectory has slowed down. In the context of the digital traces of the tourists moving in cities, these indicate places they have visited. We use spatial clustering along the path of the trajectory to find stay points, but other methods exist such as detecting trajectory segments of low movement speed.

- *Sequence-aware recommendation* is the approach to recommendation where both the previous sequence of the items the user has interacted with, and the sequence of items to be recommended are important.

- *Session-aware and session-based recommendation* is the task of recommendation where the system may have very limited information about the user (session-based) or makes a distinction between long-term and short-term history of the user (session-aware).

Problem Statement

Place recommendation relies on two main components: knowledge engineering and the recommendation model. In the knowledge engineering stage, data is collected and processed so that it represents our knowledge about the places to recommend and how these places relate to the users. The recommendation model is then built using the processed data. While the design and evaluation of recommendation models is a staple of recommender system literature, the quality of the output of the recommenders is dependent on the quality of the data produced in the knowledge engineering stage. There is currently a lack of quantitative evaluation and comparative study of data preparation methods, and the most common methods used in data preparation ignore the geographical features of the urban space.

In this thesis we focus on knowledge engineering in sequential and session-based place recommendation systems. To make behavioral models it is necessary to obtain sufficient amount of data both on individual level so that sequences of actions can be found, and in total quantity, so that we have a good representative sample of tourist behavior. The data should be expected to contain errors, which must be corrected or removed. Finally, we need an interpretation of actions on a certain level of abstraction, meaning that we describe the behavior as sequences of place visits. We examine the following questions:

- Where do we get the data about the places to recommend and how users interact with these places? What are the strengths and weaknesses of using various geo-tagged data available from public sources for this purpose?

- Assuming geo-tagged photos as the data source, how reliable is this data? What are the potential ways that the data can misrepresent the actual behavior of the users and what causes these errors? How accurately can we detect and filter incorrect values in geo-tagged photo traces and does filtering the data improve the predictive performance of the recommender?

- To transform movement traces to sequences of visits, we need to add place semantics to trajectories. The conventional method is to use a database of known POIs and infer place visits by finding POIs close to the locations that the user spent some time in. How accurate is this semantic enrichment method?

- The convention of representing the geographical location of POIs as a single point is not well suited for inferring place visits, because real-world objects have different shapes and sizes. How to represent the spatial extent and relations of places in the urban space, how can this information be automatically discovered and does this representation of places improve the quality of semantic enrichment?

- What is the inherent predictability of place visiting sequences from information theoretical perspective and how is this affected by the data preparation methods?

Several of these questions have not been thoroughly studied in the existing literature. The performance and impact of filtering errors in geo-tagged photos, when described at all, has not been quantitatively evaluated in the place recommendation context. While describing urban space as regions of interest (ROIs) is not a novel concept, the accuracy and impact of this approach in semantic annotation of movement traces has not been evaluated compared to the prevalent approach of using POI databases. We are used to seeing quantitative evaluations of predictive recommendation models, but there has been no systematic approach to evaluate the data itself to better interpret the model performance.

Contribution of the Thesis

We develop a set of methods for building behavioral knowledge used in recommending places and trip itineraries. The approach allows mostly unsupervised extraction of data from open sources. We specifically focus on geo-tagged photos which provide a source for worldwide discovery of places and visiting behavior.

The thesis makes the following contributions:

- We assess the applicability of using geo-tagged photos as the source of behavioral data by analyzing different types of open source data qualitatively. We review the literature to compare geo-tagged social media (photo sites, Twitter, Foursquare), GPS trajectories and mobile phone data in terms of spatial and temporal accuracy, sparsity and ability to represent visiting behavior. We summarize the results of our research paper presenting the Sightsmap.com application that successfully integrated open source data to map visiting behavior worldwide and used geo-tagged photos as the primary source (Chapter 1).

- We describe the two main types of errors in location or sequence of place visits in geo-tagged photos. We develop simple heuristics to filter out both types of errors and measure the effect of filtering on a synthetic ground truth dataset (Chapter 2.2). We additionally evaluate the impact of the filtering heuristics on recommender model training data preparation by comparing the accuracy of models trained with filtered and unfiltered data (Chapter 4). There are two outcomes: we show that the method we developed is effective in filtering individual incorrect locations, but we also show that our filtering heuristic is insufficient to have a positive impact on recommender model accuracy and needs to be improved further.

- We develop a method to extract stay points, or trajectory segments that constitute a visit to one place, in a way that is aware of the geographical features that direct and restrict movement (Chapter 2.3). We evaluate the accuracy of the method on small scale and find that it is over 95% accurate and of the remaining errors, 4% are correctable by a simple post-processing step. There are currently no comparable results in the literature, but our contribution has a qualitative advantage (geography awareness) and establishes the baseline for accuracy.

- We develop a new set of methods for semantic enrichment of tourist trajectories with hierarchical regions of interest (HROI), including:
  - A new method to discover the spatial extent and hierarchical relations of places with semantic and geographical awareness (Chapter 3.2).
  - A new method to create cluster hierarchies with an arbitrary number of levels by extending an existing density based hierarchical clustering algorithm.

Our new method is general purpose and adapts to local density variation (Chapter 3.3).

  – A semi-automatic method to assign place names to hierarchical ROIs by using latent space embedding to measure similarities between themes mentioned in photo captions and tags; and candidate names that come from known POIs or are synthesized from photo captions (Chapter 3.5).

- By using our proposed HROI approach, we annotate trajectories semantically by adding place names and measure the impact of this method, compared to using an open source POI database and a baseline association method. To measure the impact, we train recommender models with data prepared using the HROI approach, and the best POI association method known to us. We then compare which set of training data results in better predictive performance of the recommender models (Chapter 4). The experiments show that the HROI method outperforms the methods used in current literature.

- We introduce a metric for recommendation datasets. We adapt information theoretic methods for symbol sequence predictability to measure the predictability of item sequences in user behavioral data. We assess the impact of our HROI semantic annotation method, compared to a baseline POI method using this predictability metric (Chapter 4.3–4.4). The HROI approach results in higher predictability of the training and testing splits, compared to the baseline.

The combination of the trajectory and stay point extraction methods, noise filtering and semantic annotation with hierarchical regions of interest (HROI) is a framework designed for extracting behavioral knowledge from geo-tagged photos. The novelty compared to existing literature is that our methods are designed to be aware of the structure of urban space by modeling places as spatial regions at different scales and taking into account the geographical features that restrict or direct movement.

We quantitatively evaluate each stage separately and by cumulative impact in preparing data for the recommendation model. To the best of our knowledge, the existing literature contains no quantitative evaluation of noise filtering and semantic annotation in the context of place recommendation. Hence we provide new knowledge about the accuracy of the currently established methods. The performance of our new methods for geography-aware stay point detection and error filtering sets baselines for evaluations in these tasks that future publications can build on. Finally, our proposed HROI method for semantic annotation outperforms the current methods both by the coverage of annotated data as well as the impact on the accuracy of recommendation models.

Related Work

Knowledge discovery from big data and recommender systems are constantly growing fields of research with a large number of publications from the last two decades. In 2007, the 1st ACM Conference on Recommender Systems (RecSys) was held. The RecSys Conference Series[3] with both academic and industry presence contributes many influential papers yearly. Other important annual series include the World Wide Web (WWW) and the ACM Special Interest Group of Computer-Human Interaction (SIGCHI) conferences, the ACM Conference on Knowledge Discovery and Data Mining (KDD) and

[3] `https://recsys.acm.org/`

ACM Conference on Information and Knowledge Management (CIKM). Geospatial data mining and tourist recommendation topics are also covered in high impact factor journals, such as *Tourism Management*, *ACM Transactions on Intelligent Systems and Technology* (TIST) and *GeoInformatica*. With the wealth of research available we only review selected papers that bear the most direct relevance to the topics of the thesis. On several occasions we refer to comprehensive survey papers for more detailed overviews of the literature.

Trajectory Extraction

Recommending itineraries using crowd-sourced data begins with collecting information about tourist mobility. We will use the term trajectory extraction to mean the discovery of information about the movement or visited places of individual users. In this thesis, the purpose is to find behavioral patterns of individual users for building predictive statistical models that can make personalized recommendations. The applications for such trajectories are much wider, related to urban planning, large-scale mobility patterns in tourism, disaster management and more. A general overview of trajectory data mining is given in [144] and [93].

Choudhury et al. describe the process of extracting trajectories from geo-tagged photos that is widely used in subsequent research. They arrange photos in time-sorted sequences called streams and split these streams at longer gaps, resulting in trajectories representing day trips. This approach requires the user names, coordinates and time stamps of photos, which were downloaded through the Flickr API[4] [23]. An example of the same technique can be found in Li et al. [64]. They extract trajectories from Panoramio[5] photos to analyze tourist flows.

With sparse trajectories, it is also possible to first annotate each data point, then collapse sub sequences that have the same annotation [96, 61, 69, 12]. Lu et al. also cluster photos before connecting trajectories, but use this to discover internal trajectories of larger destinations like the Forbidden City in China [76].

With location-based social networks (LBSN), trajectories are formed from sequences of check-ins. The semantic information can be extracted directly, without the need for a separate annotation step. Noulas, et al. extract a wide range of mobility features from Foursquare data where transitions between places make up just one feature. The rich metadata of places also allows separately distinguishing activity transitions [97]. LBSNs both as a data source and as an application domain are surveyed by Zhao et al. [142].

GPS trajectories as a source of individual mobility data were investigated in the GeoLife project of Microsoft Research Asia. The GPS recordings of the movements of volunteers were used for developing techniques of extracting behavioral models. The central concept is the stay point which is detected by a sufficiently long stay in a predefined radius. Zheng et al. give a description of the stay points in the context of location recommendation in [145]. A similar technique is used by Kulkarni et al., except that they use spatio-temporal density based clustering to form regions of arbitrary size [59]. Lima et al. use GPS traces to study individual route choices, so trajectories are treated as routes between two significant locations (for example, home and work). They discover significant locations by clustering end points of trajectories and routes by clustering trajectories aligned with dynamic time warping [70].

Instead of individual trajectories, it is also possible to directly extract frequent

[4] `https://www.flickr.com/services/api/`

[5] Website closed in 2016

behavioral patterns. The T-pattern mining (TPM) method was introduced by Giannotti et al. [39] and finds similar fragments of trajectories that pass through the same regions of interest (ROI). This approach was adopted to Flickr photos by Cai et al. [13].

Going even further from raw trajectories, Jiang et al. derive a generative probabilistic model from cell phone data, where the parameters of the model are latent features of individual behavior. The model can then be used to infer individual trajectories. [56]

Some techniques for reducing the errors in trajectory extraction have been described in prior work. Invalid time stamps may be detected in Flickr photos by comparing them to the upload dates [23, 89]. Lim et al. use prior information about geolocation accuracy [69].

Among applications that require classifying the users, we look at cases where the behavior of tourists is extracted. The span of time that the user was in a given city or region can be used to classify users into tourists and residents. By taking the oldest and newest time stamps of user photos in a bounding box, the duration of their stay can be inferred [23, 40]. Flickr users can also disclose their home city and country to other users [40, 124].

Semantic Annotation

Trajectories that are sequences of coordinate and time stamp pairs are not directly usable in behavioral models. The activities or visited locations must be added to the trajectories. Such trajectories are called semantically annotated or enriched. A review of semantic trajectories topics is given in [99].

Some types of annotations are possible using only the trajectory data. GPS trajectories can be segmented to identify locations of interest ("stay points" or "stops") [4, 98]. In some applications, such as law enforcement and wildlife behavior studies, determining only the geographical coordinates where a moving object has stopped can be sufficient.

For place recommendation, visited locations need to be identified by name as distinct, human-recognizable places. The simplest method of annotation of place visits is to associate the stay points on trajectories with the closest point of interest [96, 23, 12, 76, 69].

Since the spatial distance from the user alone cannot accurately represent the probability that a place was visited, Furletti et al. expand this by annotating activities with a "gravity" model. They use a model where the popularity of a POI determines the probability that it was visited by the user [35]. Maeda et al. use a similar model to estimate place transition probability in mobile phone traces [86].

While POI databases are available, independent methods of automated POI discovery are still important. Pre-existing POI data can be integrated with application-specific relevance data, such as popularity in tourist applications, or automatically discovered attributes, like the type of activity associated with the POI. Crandall et al. demonstrated that popular places can be discovered by spatial clustering. They applied mean-shift clustering [26], which has also been adopted in several later studies that use geo-tagged photos [76, 61, 55] and Twitter [87].

In addition to working around the ambiguity caused by spatially representing places as points, it is also possible to use a more descriptive spatial representation. Regions of interest (ROIs) represent the spatial extent of places as polygons. Annotating with ROIs can be accomplished by finding spatial overlap of the stay point and a ROI [130]. The visit of a place can also be defined as the intersection of a trajectory and a ROI with a long enough duration [4]. To the best of our knowledge, there are no public databases with wide geographical coverage that would describe the spatial extent of places relevant to tourists, so methods that use ROIs also require automatic ROI discovery.

Giannotti et al. introduced the TPM method [39] for extracting regions from trajectories. Their method matches regions to known POIs. The regions that do not match known POIs are interpreted as new knowledge about places that are frequently visited. The method of Giannotti et al. has been enhanced later, for example by improving ROI boundary detection by pre-filtering stay points with low local density [20].

Extracting interesting regions by applying density-based clustering to individual stay points in GPS traces has been proposed in several papers [66, 16]. The regions are represented as a set of stay points which effectively correspond to an arbitrary shape. Uddin et al. describe an efficient method to find regions as the set of points that have many trajectories of slowly moving objects passing nearby [119].

Discovery of ROIs from geo-tagged photos has been mainly explored in the context of sightseeing recommendations. Kisilevich et al. adapted DBSCAN [32] specifically for this purpose, by defining density as the minimum number of distinct users in the neighborhood and introducing adaptive density threshold for splitting high density clusters by local variations. The modified algorithm is called P-DBSCAN [57] and is one of the more popular methods of ROI discovery [129, 89, 124].

Liu et al. modify density based clustering by introducing a predefined order in which photos are processed, so that points are assigned preferentially to clusters where there are also more popular Foursquare venues nearby [73]. Laptev et al. proposed a grid-based method, related to the Gaussian smoothing and watershed segmentation image processing techniques [62]. Their method takes the desired region size as a parameter and automatically adjusts the kernel bandwidth used in Gaussian density estimation to produce the clustering. Cai et al. use a grid-based method where the density of cells is defined as the number of intersecting trajectories [14]. Shirai et al. include the angle of view and orientation in extended photo metadata to infer the shapes of places [108]. Kulkarni et al. use online updating in a real-time application scenario and propose a method based on combining stay points that can capture the evolution of ROIs in time [59].

The discovery of functional regions [135] is usually not connected to tourism recommendation, but the applications are similar enough to consider the techniques in ROI discovery. For example, by using topic modeling, the regions are characterized by the latent properties of the places that reside within the region. This can be used to combine individual places into wider level regions [36, 135].

The POI discovery method of Brilhante et al. can also produce regions with high place density. They query geo-referenced Wikipedia pages within a given bounding box and apply density-based clustering to group objects that are closely together [12].

Along with our ROI discovery method we also introduce a method of extracting a cluster hierarchy with a given number of layers. General purpose methods that select significant clusters while maintaining the hierarchy relation have not been studied extensively. Sander et al. present a method that recursively splits the points to clusters at significant local maxima of a reachability plot. Trees formed by conventional hierarchical clustering are handled by converting them to reachability plots [106]. Campello et al. describe both a simplification of the cluster tree by setting a minimum cluster size and an optimal method of creating a flat clustering based on cluster stability [15]. By functionality, the most similar method to our work is the Auto-HDS algorithm by Gupta et al.[43], with results equivalent to sampling of the cluster tree [15].

A hierarchical model of the urban space has previously been studied in the GeoLife project. The GeoLife recommender uses a hierarchical model to discover similarities between users and recommend friends and locations. The hierarchy is created by iterations of density based clustering [66, 145].

When using automatically discovered ROIs, the problem of finding the semantics of the ROIs has to be solved. Crandall et al. showed that for a place represented by a set of photos, simply selecting the most distinctive tags can generate accurate labels [26]. Yin et al. apply a generative mixture model using the sets of Flickr photo tags in a location and the overall distribution of tags to find the most likely names [131]. Majid et al. annotate ROIs using textual tags from photos and POI names from Google Places [89]. For POI names, catalogs of POIs, or gazetteers, have been used [76, 55].

In addition to place name annotations, trajectories may be enriched in other ways that can enhance recommendation. Arase et al. use a TF/IDF approach to find the theme of each trip after extracting and annotating the trajectories [8]. Liao et al. combine geographical data and temporal parameters of trajectories in a probabilistic model to infer activities performed at visited places [67]. Skoumas et al. use natural language processing to generate a knowledge base of POIs that includes their relations (e.g. "in", "near") [109]. Falher et al. introduce a method of describing regions in terms of similarity to other, known regions. This can be used to recommend new regions to users based on their previous activities [34]. Quercia et al. collected crowd-sourced ratings for street scenes to evaluate aesthetics of routes. They complement this with a method to scale up by rating new places by their similarity to the human-rated ones [102]. Zhao et al. introduce a method to associate tweets with POIs to annotate them with dynamic and real-time information [141].

Semantic annotation may be performed interactively using visual analytics tools [5, 6] that assist the user by clustering closely located places and displaying distributions of POI types and visit times. The user assigns the semantic label to the ROIs and trajectories are annotated automatically by finding ROIs that the stay points intersect.

Sequence Prediction and Recommendation

We view the place recommendation as a sequence-aware recommendation problem. The recommendations should take into account the previous places that the user has visited and when several places are being recommended, they should fit together and be ordered in the way that best serves the user. Quadrana et al. review the applications and methods of sequential recommendation [101].

We first briefly review the literature of trajectory-based location prediction. Similarly to place recommendation, it deals with modeling or predicting the "intent" of users. Giannotti et al. introduced the formalism of temporally annotated sequences [38], which has been used in later research for frequent pattern mining [13] and prediction [95]. Several techniques to improve location prediction have been proposed in this context that could also have uses in next place recommendation. Grouping of similar objects can improve detecting common behavior [137]. With the assumption that related activities happen close in time, clustering of events to minimize intra-cluster time differences can be used to improve pattern discovery [72]. Liu et al. study sequential decisions from the point of view of learning the reward functions that motivate the decisions [74].

There have been by two research directions in itinerary recommendation that until recently had little overlap. In operations research literature, the problem has been treated as one of combinatorial optimization, related to logistics problems such as vehicle routing. In the mainstream of recommender systems publications, the focus has been on behavior modeling and knowledge discovery.

An influential paper by Vansteenwegen and Van Oudheusden [122] formulated the tourist trip design problem (TTDP) as an operations research opportunity, becoming a foundation to the body of research that models trip recommendation as a generalization of the orienteering problem (OP). The generalizations extend the problem with various constraints, such as multi-day visits and time windows, but also max-n type constraints [42]. Recently, TTDP has been studied in the group recommendation setting [46]. An overview of the TTDP literature is given in [37].

The orienteering problem (OP) representation is also applicable to machine learning based approaches to recommendation. In this case, the extracted routes are used to make a connectivity graph which is then used to search for itineraries [23, 69]. The TripCover problem which is similar to the OP is formulated in [12]. The solution is a set of trajectory fragments which is then arranged into an optimal sequence.

Treating itinerary and place recommendation as a sequence prediction problem is the other major direction. Recommenders that follow this paradigm often rely on mining crowd-sourced data and using machine learning to build recommendation models. Because other sequential prediction tasks, like music playlist recommendation or predicting the next visited location are similar, we include papers from these application domains in the review.

Probabilistic approaches learn generative hierarchical probability models, usually through maximum likelihood estimation. Latent topics are used to capture the taste or preference of the users. Kurashima et al. build a probabilistic model that predicts the likelihood of a transition to a place. They combine a 1st order Markov model with a personalizing topic model [61]. An updated version called the Geo Topic model was designed to improve the personalization of the recommendations [60]. Probabilistic models can be enhanced by including temporal [75], activity category [107] and context information [129, 143].

Latent space embedding is a related approach, popularized by Chen et al. They fit a dataset of song transitions into a space of latent features. The distance in this space represents transition probabilities between any combination of songs, overcoming sparsity of training datasets [19]. Zhang et al. use a joint embedding of location, time and text content of geo-tagged social media posts to predict locations and activities of users [138].

Matrix factorization methods aim to solve typical problems in recommendation, data sparsity and generalization. In the simplest form, they represent both users and items in a latent, lower dimensional space and the preference for items is found by the dot product of user and item vectors. Since this method relies on user and item similarities to capture preference, it is a form of collaborative filtering (CF). Matrix factorization was adapted to a sequential model by Rendle et al. in predicting the next action of the user based on recent actions [103]. The Factorized Personal Markov Chains, or FPMC model of Rendle et al. has inspired further adaptations [21].

General purpose supervised learning models have been successfully applied in sequential recommendation. Noulas et al. describe a supervised learning approach to predicting users' next venues. The learning features include behavioral history and spatial characteristics. They achieve best results with decision trees [97]. Muntean et al. model next place prediction as a learning to rank problem, using support vector machine and gradient boosted regression tree methods. They rely on a rich feature set extracted from geo-tagged photos and Wikipedia [96]. Wang et al. propose a two-layer hierarchical neural network structure for next-basket recommendation [126].

With the rising popularity and successful applications of deep learning, various artificial neural network (ANN) approaches have been proposed. Recurrent neural networks are particularly well suited to learn sequential relations and are the basis of the GRU4REC method of Hidasi et al. [47]. Attention mechanisms in neural networks were successfully employed by Yin et al. [132]. Relevant ANN methods are too numerous to list here explicitly so we refer the reader to a recent survey [139].

Neighborhood-based methods have received attention recently in sequential and session-based settings [84]. k-nearest neighbors methods adapted to sequential recommendation were proposed by Jannach et al. [49] and a similar technique based on playlist similarity by Turrin et al. [118].

A large number of publications in recent research focus on the predictive performance of recommendation models. Because sequential and session-based recommendation is a quickly developing direction of research, concerns about neglect of practices that promote unbiased evaluation and reproducible research have been raised [83, 28]. For example, Rendle et al. argue that method tuning plays a major role in benchmark results [104]. Recommendation methods should therefore currently not be judged only by reported performance evaluations, but by their general suitability for a given application domain.

We also examine the sequential recommendation task from the perspective of the predictability of behavior. Predictability of human movement based on the Shannon entropy of sequences was investigated by Song et al. [112]. Their method has been the subject of later publications that offer additional proofs, clarifications and experimental validation [110, 128]. Li et al. use similar techniques to estimate the lower and upper bounds of the predictability of check-in behavior of LBSN users [65].

Human-Centric Aspects of Recommendation

In the thesis we have taken the simplified approach of using the prediction accuracy as a proxy measure of expected user satisfaction. There is a significant body of research that deals with a more user-centric approach to the quality of recommendations. Mostly, the discussion involves the closely connected topics of recommender system optimization and evaluation.

The traditional accuracy metrics do not always distinguish between significant differences in recommendations between different algorithms. For example, accuracy may be improved by bias towards popular items, but this is not always the desirable result [94, 50]. Metrics for measuring diversity, as well as novelty, are given in [123, 48]. Measuring the satisfaction with a sequence of items is described by Masthoff and Gatt [92].

In recommender systems research it is recognized that recommending a sequence of items is a different problem than rating single items [91]. Adamopoulos suggests that in recommendation of sets of items, interaction between individual items plays an important role [1]. Hansen and Golbeck [44] define main criteria for evaluating sequences: individual item ratings, order interaction effects and co-occurrence interaction effects. Their work does not formulate any solutions, but an example conforming to the above criteria is given in [22], where collaborative filtering (CF) is used to recommend item sequences. Interaction effects are further studied in [125].

A fundamental problem in recommending sets of items is the dilemma of accuracy versus diversity [111, 134, 146, 58, 24]. The recommended items should by dissimilar to each other and at the same time relevant for the user. Diversity can be considered a co-occurrence interaction effect, falling under Hansen and Golbeck's three criteria.

Table 1: Comparison of recommenders by type, knowledge sources and sequential recommendation criteria

Reference	*Type*	*Pers.*	*Behavior source*	*POI source*	*Seq. aware*	*Sess. aware*
Choudhury et al.[23]	itinerary	no	Flickr	Lonely Planet	yes	no
Kurashima et al.[61]	itinerary	yes	Flickr	Flickr[26]	yes	yes
Lu et al.[76]	itinerary	yes	Panoramio, travelogues	gazetteer (unspecified)	yes	yes
Brilhante et al.[12]	itinerary	yes	Flickr	Wikipedia	yes	no
Lim et al.[69]	itinerary	yes	Flickr	Wikipedia	yes	yes
Chen et al.[18]	itinerary	yes	GPS traces	Foursquare	yes	yes
Cheng et al.[21]	next place	yes	Foursquare Gowalla	Foursquare Gowalla	yes	yes
Liu et al.[75]	next place	yes	Foursquare Gowalla	Foursquare Gowalla	yes	yes
Majid et al.[89]	top-n	yes	Flickr	Google Places	no	no
Zheng et al.[145]	top-n	yes	GPS traces	unspecified	no	no

The concept of item list diversity was introduced by [147].

In the context of route or itinerary recommendation, some papers have explored these topics. Yin et al. include diversification of patterns in the context of ranking extracted patterns [131].

Recommender Systems

Several complete tourist recommender solutions that make use of the geo-tagged photos of Flickr [23, 61, 12, 69, 89] and Panoramio [76] have been proposed. Recommenders that use GPS traces [18, 145] share some similarities, as they are also required to solve the tasks of trajectory extraction and semantic enrichment. Another relevant class is the location-based social network(LBSN) next place recommenders [18, 21, 75]. They represent sequence-aware systems that deal with sparse check-in histories similar to photo traces.

Table 1 classifies the recommender systems from the sequence recommendation and knowledge discovery perspective. *Type* describes the type of output that the recommender produces (a single place, a trip itinerary or top-n locations). The recommender is *personalized* when it can produce recommendations tailored to the specific user. *Behavior source* describes the source of behavioral data for creating the recommendation model. *POI source* gives the provider or method of place data. *Sequence aware* recommenders fit their recommendation into a prior sequence of items or give a recommendation that is a coherent sequence. *Session aware* recommenders tailor the recommendation to current session. We have extended this category to include recommenders that fill itineraries between the designated end and start items, or with mandatory place visits.

The most common method of recommendation in the papers we review is representing the transitions between places as a graph. Rewards based on matching with the users' preferences or overall popularity are associated with the vertices of the graph. The recommendation is computed by finding the most rewarding path on the graph within given constraints [76, 23, 12, 69]. Kurashima et al. [61] build a probabilistic model that predicts the likelihood of a transition to a place. They combine a 1st order Markov model with a personalizing topic model. An updated version called the Geo Topic model was designed to improve the personalization of the recommendations [60].

FPMC-LR of Cheng et al. uses a matrix factorization method where recommendations are personalized using collaborative filtering (CF), leveraging the behavior of similar users [21]. A similar approach to address the sparsity of user-item interactions is taken by Liu et al. [75] who construct a low-rank graph representation of the interactions through a hierarchical probabilistic model. Their model also captures evolving preferences.

Individual place recommendations are similar to the problem of item recommendations in e-commerce or entertainment and k-nearest neighbors collaborative filtering model has been proposed by multiple authors [145, 89]. Majid et al. describe a context-aware recommender, where the set of places available for recommendation is pre-filtered using the current context [89]. Zheng et al. additionally recommend friends. While they use sequences in similarity calculation, the recommendations are not sequence-aware [145].

New trip and next place recommendation approaches are being proposed constantly. Additional information and references can be found in a more detailed survey on recent advances, with a focus on geo-referenced social media, compiled by Lim et al. [68].

Study of Tourist Recommenders in our Research Group

Prior work on tourist recommenders by our research group is based on content-based filtering, where recommended items are matched to user profiles by their similarity to user interests. Tammet et al. outlined the architecture of the rule-based recommender, where knowledge about items and users is assigned a confidence rating that expresses the reliability of the information. Knowledge is integrated from online sources and new knowledge can be deducted through inference rules [82]. The item ranking and itinerary planning algorithms were further detailed by Luberg et al. [81]. In this paper we also described the in-memory database architecture used for inference, item ranking and planning tasks.

Subsequent work concentrated on creating and maintaining the knowledge base with sufficient quality for practical applications. When integrating data from multiple sources, repeated occurrences of the same item need to be recognized and merged [78, 80]. A machine learning approach to increase the accuracy of item deduplication was introduced by Luberg et al. [77]. The recommender system must also derive object properties from sources in different languages and store object descriptions for a multi-language interface [78, 80].

The rule-based architecture and the knowledge discovery techniques were used to implement a prototype tourist recommender system for Tallinn and Riga. We also developed a trip planning application for the national tourism promotion website, Visit Estonia[6].

[6] `http://www.visitestonia.com`

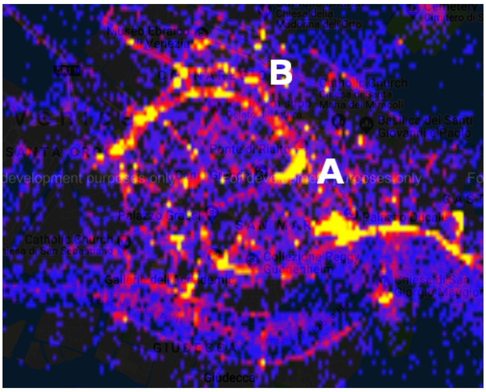

Figure 2: sightsmap.com heat map of Venice. The reverse "S" shape of the Grand Canal is clearly visible (A-Rialto bridge). A scenic walking route into the Cannaregio district that many tourists arriving in Venice take is marked with B. Map data by Google.

1 Destination Discovery from Geo-Tagged Photos

To build a recommendation model, we are interested in two types of data: the items to recommend and the behavior of users regarding those items. In destination recommendation the items are points of interest (POIs) which the users then visit. The sequence of visits is relevant in trip itinerary recommendation, both logistically and in terms of the themes and content of the trip.

While there are publicly accessible databases of points of interest, such as Google Places, information about the travel behavior of individuals is not directly available in sufficient amount to build accurate models. We therefore have to extract this implicit information from other data the users have shared publicly. In this thesis we focus on geo-tagged photos which can be used to extract both behavioral data (user-item interactions) and find destinations (items) to recommend.

1.1 Geo-Tagged Photos as an Ubiquitous Data Source

The sightsmap.com[7] application [114] demonstrated that it is possible to build a worldwide database of places from crowd-sourced data. The primary focus of the application was to browse hot-spots of interest in any region using an interactive map.

The map displayed two types of information: the heat map representing the geography of tourist visits; and the individual places including their names, photos and popularity ranks. Users were able to zoom in to view higher resolution heat maps (Figure 2) and browse places from the worldwide scale to city block scale. Through integrating multiple sources and analyzing textual metadata of images, the application was also able to categorize places (Figure 3).

The application was built using photos from Panoramio (now closed). Using the geographic location of each photo, spatial density of photos taken implicitly show the popularity of locations. The spatial density of the photos was used both for creating the heat maps and discovering individual locations. By clustering hotspots on the heat map and matching them with nearby places from Wikipedia and Foursquare, the application linked the hotspots to geographical objects and sightseeing places.

[7] `http://sightsmap.com`

Figure 3: sightsmap.com places matching the filter "beach" in Northern Estonia. Map data by Google.

In the context of knowledge discovery for destination recommendation, the most important categories of data are the names and popularity of places. The sightsmap.com application showed that geo-tagged photos can provide:

- names and locations of the most visited places;
- spatial distribution of tourist interest, including areas between and around points of interest;
- data integration with heterogeneous sources;
- worldwide coverage.

The most obvious benefit for this approach is that destination data for almost anywhere in the world can be obtained using the same process and automated tools.

Panoramio photos were perfectly suited for this application for two reasons: first, they were geo-tagged as accurately as possible and second, the primary focus of the site was to provide interesting photos of places and scenery. The users uploaded the photos that they considered worthwhile for other users to browse. Each photo can be therefore be considered as an implicit recommendation of the specific place that was photographed.

Besides being able to cover a particular geographical area, how well the method performs within the given area is also important. We consider two metrics: 1.) how many out of the notable points of interest in an area does the method discover; and 2.) out of places that the method has associated with the names, how many have correct names.

We performed an experiment with the sightsmap.com data from the United Kingdom and France to measure how many notable places were discovered. For both countries, 56% of places with a Wikipedia entry were found. The total number of places with Wikipedia entries was approximately 10000 in both cases, with more than 5000 places correctly located [114]. While these places, by definition, all have correct names, the experiment did not measure the accuracy of name association over all of the hotspots that the method identified. We will address the accuracy of place name association in Chapters 2–3.

With the use of geo-tagged photos as data source also come specific problems related to data availability, privacy and bias. The closure of Panoramio in 2016 is a prime example that the business interests of data provider companies are the major factor in deciding whether open data platforms are developed and maintained for public use. The use of such commercial services then has risks when planning sustainable applications.

Data licenses of currently existing providers like Flickr and Foursquare have requirements that their data is not retained for a long period of time. This effectively prevents distributing benchmark datasets that are based on the data from those sources, which limits the reproducibility of studies done using the data. By terms of use, users of these platforms have the right to have their data deleted, which would be impossible in case the data is openly redistributed. Furthermore, EU legislation considers location and photo data personal, meaning that geo-tagged photos fall under data protection laws even if the data does not contain, for example, names or e-mail addresses of the users. Under EU legislation the users also have the right to object to processing of their data for scientific research.

Finally, recommender systems should strive towards unbiased representation of their user base, while social media is well understood to have demographic bias. For example, Hecht and Stephens [45] identify demographic bias towards urban users in various geo-tagged information and bias towards urban areas which affects the quality of coverage.

1.2 Alternative Data Sources

There are several alternatives to geo-tagged photos that are suitable for mining destinations and travel behavior:

- location based social networks (LBSN);
- geo-tagged social media posts;
- mobile phone positioning data and call records;
- card payments or validations;
- recorded movement trajectories.

Location based social networks (LBSN) can be used as sources of travel destination information. For example, Foursquare can provide coordinates, names, fine-grained categories and popularities of places worldwide. Therefore it is a viable alternative to geo-tagged photos for finding destinations to recommend. Foursquare does not allow access to user's individual histories and cannot be directly used to infer a personalized behavioral model. The users may "leak" their check-ins intentionally via Twitter which can produce sufficient volume of data for some modeling purposes [97, 17, 100]. Another issue that affects the ability to extract behavioral information is that the incentives offered by the social media platform cause the users to misrepresent their behavior [140].

Twitter is the primary example of social media where posts optionally include geo-tags. Tweets that have precise location geo-tags have been associated with known POIs with up to 60% accuracy [141]. Given that individual user histories can be extracted from Twitter, it is a potential source for developing behavioral models. Similarly to other platforms, where users share information through social use of applications, geo-tagged tweets represent a biased sample of the population [90, 105]. Many empirical studies on geo-tagged tweets use data from before April 2015 [25, 87, 138, 3]. From April 2015, Twitter changed its policy regarding precise location geo-tags to opt-in instead of opt-out, drastically reducing the ratio of posts having geo-tags [31]. Similarly to geo-tagged photos, the sparsity in time needs to be taken into account when extracting personal trajectories from Twitter.

Mobile phone data is typically available to telecom providers. Excluding applications developed by the telecom providers themselves, use of the data would require some incentive for the data provider while not violating the legislative measures and public trust related to privacy [2]. Some providers make aggregate, such as k-anonymity protected data available as a paid service. This type of data is suitable for discovering destinations but not for personalized models, as the preferences of several people are aggregated.

Mobile phone positioning data is temporally sparse. The position of the phone is recorded when the user actively uses the phone or moves from the range of one cell tower to another. The network may periodically record the position of the phone, with the typical period being about 30 minutes to one hour [127]. The spatial resolution depends on the distance between the cell towers, which can be as low as 50m in dense urban areas [54]. Unlike LBSN check-ins and to a lesser extent, sightseeing photos, the locations of positioning events are not related to specific places but rather to call events and the geographical placement of cell towers.

Smart card transactions, such as the RFID cards commonly used in public transportation fare collection provide individually traceable human mobility records [88]. Since the card transactions normally take place when the user boards or exits a transport service, their sparsity and connection to infrastructure rather than points of interest makes them less applicable to destination mining.

In contrast to the previously mentioned data, recorded movement trajectories using technologies such as GPS provide both very high temporal resolution and high spatial resolution. Users normally do not record their daily movements and there are few datasets with substantial number of trajectories available. The well-known examples include the GeoLife dataset of the daily activities of 167 volunteers, mostly based in Beijing [145, 133] and the New York City taxi dataset [30]. Well-designed applications can however generate enough data to cover smaller areas [120].

2 Trajectory Mining from Geo-Tagged Photos

The first step in making a predictive statistical model for place recommendation is to extract the trajectories of tourists from the data. We begin with a set of photos, where each photo has the following attributes:

1. user identifier;
2. geographical coordinates;
3. time stamp;
4. image caption.

In Section 2.1, we describe how to extract the movement trajectories of individuals from the set of photos and to separate tourists from residents. The trajectories are initially noisy, both due to users entering invalid data manually and because of incompatibilities and inconsistencies in the way digital photos are transferred from the camera to the database of the social photo platform. We describe common types of noise and how to cope with these types in Section 2.2. Detecting place visits, also called "stay points", is described in Section 2.3.

We evaluate the methods described in this chapter on photos from Panoramio. Because the ground truth about the location and time of the Panoramio photos is not available, we test the efficiency of the noise filtering on a synthetic dataset (Section 2.4). The synthetic data is created by sampling distributions estimated from a small manually verified subset of the real world data. We also report the statistical results of applying the preprocessing described in this chapter on real world datasets.

2.1 Extraction of Raw Trajectories

The extraction of trajectories of tourists follows the commonly used heuristics introduced by Choudhury et al. [23]

1. We group photos by user and sort them by time stamps, producing photo streams. With Flickr, API data is sufficient. Panoramio API did not provide time stamps, so EXIF metadata had to be scraped separately from the website and integrated with the API data.

2. We filter the photo streams to remove users that behave differently from tourists. This is accomplished by examining the length of stay period in the area and removing users who stay longer than 15 days. We also remove streams consisting of only one photo.

3. We split photo streams into sub-sequences at places where a gap between two photos is 8 hours or more.

This process produces sequences of photos, typically taken within one day, where each photo has the contextual attributes of time and geographical coordinates. We will refer to these sequences as trajectories of tourists.

2.2 Noise Filtering

Public geo-tagged photos as a data source have distinct features that call for methods specifically adapted for the problem. While the resolution of the spatial coordinates is comparable to GPS traces, the data is noisier due to altered coordinates and time stamps. The data is also much sparser and irregular.

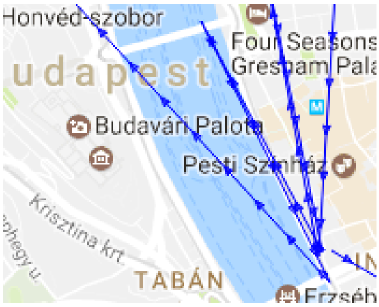

Figure 4: Time stamp noise. Erratic trajectory resulting from original time stamps being overwritten. Map data by Google.

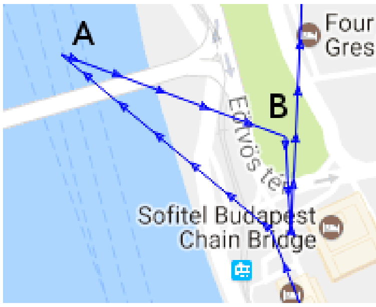

Figure 5: Coordinate noise. Invalid coordinates of photograph A can be detected here because the time interval between A and B is only 26 seconds. The likely cause is the user taking a photo of the bridge from the park and assigning the coordinates of the bridge to the photo A. Map data by Google.

The location and the time a photo was taken is present in the auxiliary metadata tags of the EXIF image format used by digital camera makers and supported by Flickr and Panoramio. This metadata is sometimes referred to as "EXIF data". Image processing software and tools provided by the photo-sharing social networks can however arbitrarily alter this information both in the EXIF format and after storing it as separate metadata.

Because of the image processing and uploading workflow, the time stamps of images can represent any of the following: the time the photo was taken, the time the photo was edited on the user's computer, or the time the photo was uploaded. The cases where the original time stamps are not preserved result in trajectories that cannot be used to reconstruct original visit histories (Figure 4). We refer to these cases as "time stamp noise".

Many cameras do not have GPS receivers. While consumer GPS also may have positioning difficulties due to signal reception that result in invalid coordinates being recorded, a much more prevalent problem is user errors in cases the user enters the coordinates manually. A case we observed frequently is that when photographing a remote object, the user enters the coordinates of the object, not where the photo was taken from (Figure 5). Users also confuse similar-looking buildings that are not major landmarks. In the thesis we refer to this as "coordinate noise".

There are two main approaches to noise filtering in trajectories: smoothing filters and removal filters. Smoothing filters can be purely statistical (mean or median filters) or model a physical process (Kalman filters) [144]. While these methods have strong theoretical foundations and are widely used, they are not appropriate for filtering photo sequences. Each photo represents a potentially legitimate place visit and displacing them through smoothing would change their semantics.

In trajectory preprocessing, ad hoc heuristics that detect movement speeds over a given threshold can be used as removal filters [144]. We have adopted this approach to develop a simple heuristic filter. We measure the speed of transition between consecutive photos. If this speed is over a given threshold, one of the two points is classified as noise. While this approach can also use statistical detection of outliers, we did not observe any improvement over a simple hard threshold.

To compute the transition speeds, we use OpenStreetMap[8] and the Open Source Routing Machine (OSRM) router [85] to find realistic distances between places in the urban environment. Bodies of water, buildings and major roads all form obstacles which make straight line distance inaccurate. In some tourist destinations, historic fortifications may cause pedestrians and vehicles both make long detours.

The OSRM router works by calculating distance along discrete objects called road segments, so it is not accurate on very small scale. For distances below 30m, we instead calculate straight line distance. The speed filter must also be desensitized to cases where photo time stamps are truncated. When photos are taken in close proximity, this can result in transition times of 0. We use additive smoothing of time intervals where the distances between photos are small.

The noise filter uses the cut-off parameter v_{cut} to detect transitions between photos where the estimated movement speed is too high. For consecutive photos i and $i+1$, if $v_{i,i+1} > v_{cut}$ then either photo i or $i+1$ could have invalid coordinates. The filter then calculates, which of the i and $i+1$ should be removed so that the new transition that is formed by the removal would have lower estimated speed. After the removal, the next filter iteration starts with the newly formed transition, so it is possible for both photos to be removed.

To detect time stamp noise, we use a heuristic of based on the notion of trajectory reliability. For each trajectory of length n, if the ratio of removed photos $\frac{n_{cut}}{n} > r_{noise}$ then the trajectory is considered unreliable and discarded. r_{noise} is the tolerance level for noise in a trajectory.

2.3 Stay Point Detection

To prepare the trajectories for semantic annotation, we need to detect locations that the user has visited. In trajectory data mining, these locations are called "stay points" [144]. Because of the sparsity of photo traces, the basis of stay point detection is that each photo represents a legitimate place visit. However, it is not reasonable to treat photos that are taken closely together as separate places. The suitable method of stay point detection is then to cluster the photos based on their spatiotemporal proximity.

The general method for spatiotemporal density clustering is ST-DBSCAN [9, 10] that uses discrete time and expands spatial neighbors in one time window with those in adjacent time windows. However, the heterogeneity of space and time dimensions and density variations are not well addressed by this approach [9].

Since the purpose of stay point detection is to partition trajectories into episodes of stops and moves, the clustering only needs to group points that follow each other in the sequence. This observation has been used in trajectory-specific methods [136] that require specifying separate time and distance thresholds for inter-cluster density.

[8] https://www.openstreetmap.org

We developed a new stay point detection method that also exploits the sequential nature of the trajectories. However, our approach allows us to both include geographical information in realistic distance estimation and addresses the different scale of space and time dimensions. We convert the trajectory points into one-dimensional space by the following procedure: the first point in the trajectory is placed at coordinate 0. The coordinate of each subsequent point i is the cumulatively increasing distance $d_i = d_{i-1} + \delta_{i-1,i}$. We use the travel distance calculated using OSRM as $\delta_{i-1,i}$. This allows us to omit the time dimension, because the routing distance reflects cases where one location is in close proximity to another, but cannot be reached directly. While street network routing distance is not metric, the resulting space after the conversion is metric and standard density-based clustering can be applied to group the photos.

2.4 Noise Filtering on Synthetic Data

The ground truth about the coordinates and times of the photos of real users is not available. To measure the accuracy of the noise filtering heuristic that we propose, we created a synthetic dataset of 9910 trajectories where the noisy photos were labeled.

We simulated movement between known POIs in Budapest to create movement trajectories. We then sampled distributions estimated from manually annotated trajectories to generate photos that would be taken when tourists move along the simulated trajectories. Finally, we created time stamp noise by changing the times of some trajectories; and coordinate noise by adding photos that simulate the scenarios of setting coordinates to a remove object and misidentifying a location. [51]

The noise filter was evaluated by the precision and recall metrics. Denoting true positives as TP and false positives as FP, precision is the ratio of real noise among the photos that are filtered out:

$$Precision = \frac{TP}{TP+FP} \tag{1}$$

Recall represents the ratio of real noise detected by the filter:

$$Recall = \frac{TP}{TP+FN} \tag{2}$$

where FN is false negatives. We measured the effect of parameters v_{cut} and r_{noise} by fixing one parameter and varying the other. The left graph in Figure 6 shows the effect of changing the cut-off speed with the fixed $r_{noise} = 0.15$. The precision of the filter has a "knee" shape with a rapid drop where the cut-off parameter comes close to the median of movement speeds in the dataset. At the same time, recall improves slowly with the decrease of the cut-off speed, reaching only 0.6 near the "knee" point. This shows that nearly 40% of noise is associated with speeds below 10 km/h.

In the right graph in Figure 6 we show the effect of the noise tolerance parameter, with $v_{cut} = 10km/h$. Surprisingly, the precision of the filter forms a similar knee shape, showing that the there is a threshold of the ratio of removed photos in a trajectory where the probability of the trajectory being "noisy" increases rapidly. However, recall is still low even where the filter is clearly too aggressive ($r_{noise} < 0.1$).

In our original publication we reported that the filter is successful in detecting coordinate noise, with 97% of the coordinate noise removed at the filter settings of $v_{cut} = 10km/h$ and $r_{noise} = 0.15$ [51]. This can be interpreted as a positive outcome. The recall was much lower when all types of noise were included (Figure 6), resulting in

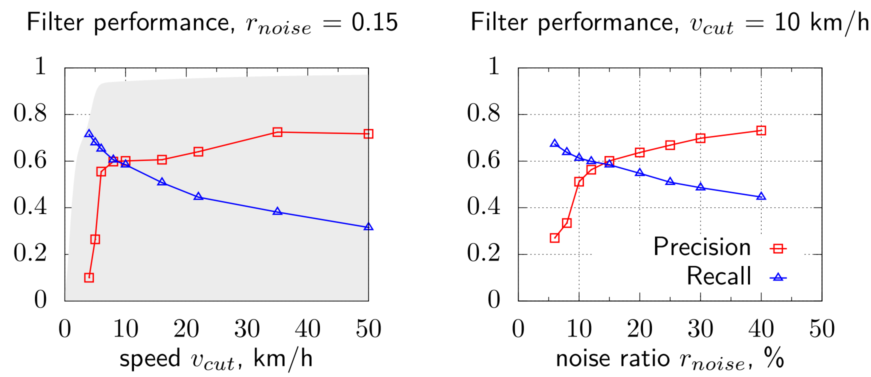

Figure 6: The performance of our proposed noise filter on the synthetic dataset. Selecting the cut-off speed (left plot) and the noise ratio threshold (right plot) involves a compromise of precision and recall. In both cases there is a sharp drop off in precision when the filter becomes too aggressive. The gray background represents the cumulative distribution of speeds in the dataset.

approximately 60% of the noise detected. Nevertheless, one type of noise was successfully removed. However, this interpretation of the result does not consider the overall low precision of the filter, which causes the removal of valid records. We investigate the effectiveness of the filter again from the perspective of its impact on the training quality in Chapter 4.

2.5 Preprocessing on Real World Datasets

We have extracted tourist trajectories in metropolitan areas that are popular tourist destinations. From Panoramio, we downloaded photos taken in Budapest, Vienna and Venice. Since the Panoramio API was closed, and to diversify the selection of destinations, we downloaded photos from Flickr for a total of 7 cities.

Table 2 reports the statistics for the destinations. The columns *source* and *city* specify the API that the photos were downloaded them and the target area. *Photos with EXIF* gives the number of photos that had sufficient metadata (location and time) available. This is relevant in case of Panoramio where the time had to be scraped separately from the website. *Users* is the total number of users in the dataset and *tourists* is the number of users classified as tourists, as described in Section 2.1. After filtering with $v_{cut} = 10km/h$ and $r_{noise} = 0.15$ and stay point detection, the number of trajectories extracted from the dataset is given in the column *trajectories*.

There are some common characteristics across all datasets in Table 2. About one third to half of users posting photos are classified as tourists. Those users contribute on average from 0.88 to 1.5 trajectories each. The most important questions regarding these results are whether they form a good representative sample and whether there is enough data to build accurate predictive models. The first question is not possible to answer without a secondary data source about tourist visits in the same region. We argue that by posting photos of scenic locations the users make implicit recommendations, which makes it suitable in place recommendation. This view has been supported e.g. by Crandall et al. [26]. However, the sample size is relatively small. While millions of tourists visit Estonia and Tallinn yearly [63], we only collected data

Table 2: Number of geo-tagged photos, tourists and movement trajectories in cities.

source	*city*	*photos with EXIF*	*users*	*tourists*	*trajectories*
Panoramio	Budapest	177496	10688	3244	3240
	Vienna	93912	7197	3237	3619
	Venice	61200	8426	3985	3488
Flickr	Tallinn	112336	3974	2073	2531
	Budapest	450967	13401	6161	8559
	Vienna	649154	14062	6688	9447
	Venice	516570	25495	13199	16198
	Tokyo	1827114	23666	8418	12800
	Los Angeles	2128551	39943	13445	14350
	Paris	1945023	61652	25765	35515

about approximately 2000 users over a period of 15 years. At the same time, mobile phone positioning data covered approximately one third of the visitors [2]. This implies that despite wide adoption of photo sharing sites, geo-tagged photos are able to cover only a small minority of visitors.

Due to the workload of manual annotation and validation, not all of the trajectory sets in Table 2 have been used in further experiments covered in Chapters 3–4. Since different experiments were performed over a period from 2016 to 2019, they use a different subset of trajectories, with early experiments being based on Panoramio data and later ones on Flickr or combining both sources.

We did a small-scale qualitative validation of the tourist classification method that was adopted from the literature and the new stay point detection method that we introduced in Section 2.3. We reviewed 303 trajectories extracted from Flickr photos taken in Budapest and 58 trajectories in Tallinn, also from Flickr photos.

To evaluate tourist classification, we counted users who did not visit at least two typical tourist attractions and whose photos explored artistic themes or were more oriented towards portraits instead of scenery. Out of 234 users, 23 or 9.8% were possibly non-tourists. This does not include an event on October 25th and 26th in 2011, when 52 one-time users participating in a T-Mobile 4G promotional game posted photos from predetermined locations in Budapest.

For clustering quality, we counted clusters that a.) should have been merged with the previous cluster; or b.) should have been split. Out of 1716 stay points on the trajectories, we found 67 (3.9%) clusters that should have been merged with others and 8 (0.4%) that included photos from obviously distinct stay points.

We analyze the results of trajectory extraction evaluation the context of using the trajectories in recommendation. Non-touristic users whose trajectories are disjoint from tourists do not contribute to recommendations. Those who have visited moderately popular hotspots, do contribute. It is not necessary to exclude this contribution from the recommendations, as it adds some local diversity and the majority of the recommendation knowledge still comes from the tourists.

Undesired splitting of single stay points to multiple clusters that was observed in 4% of clusters causes apparent repeated visits to the same place or area. This introduces some bias in the recommender towards predicting the same place multiple times in a row, which is not useful in typical tourist recommendations. However, this effect is easy to work around after the annotation stage by merging subsequent stay points with the same annotation.

3 Place Identification and Trajectory Annotation

Semantic annotation of trajectories has various applications, such as predicting the route of moving objects and social network discovery [93]. It is also an important intermediate step in transforming the histories of users in raw form (GPS traces, sequences of geo-tagged photos or other forms of trajectories) to sequences of place visits that can be used as an input in building a recommendation model.

In this chapter we assume that the source data comes from geo-tagged photos, preprocessed as described in Chapter 2. We begin with trajectories where stay points have been annotated by photo clustering, but do not have names. The goal is to identify which places each stop corresponds to and annotate the stops with place names.

This task is commonly accomplished by using a database of POIs and associating each stop with a nearby POI [96, 23, 12, 76, 69]. In the common case where there are several nearby POIs, the heuristic of choosing the correct one can be as simple as selecting the nearest, or using some weight function that expresses preference towards places that the user is more likely to visit (such as popularity or place category). We present an experiment with a popularity-based heuristic that we developed in Section 3.1.

While the heuristics of POI selection can improve accuracy by integrating additional information, the POIs themselves are represented as pinpoint coordinates. Figure 8 illustrates how the extent and shape of an object can complicate associating photos with POIs. The coordinates of the football stadium in a POI database are located in the middle of the pitch, while the photos are mostly taken from the stands.

We propose that instead of POIs, the photos taken by tourists, possibly augmented from other sources like Flickr, should be mined for regions of interest (ROIs). The regions are defined by their geographical boundaries and are represented by closed polygons on the map. Such regions have hierarchical structure, with larger areas (a park, an island) embedding smaller places (a museum, a chapel).

In Section 3.2 we describe the method of extracting such regions from geo-tagged photos using density-based clustering. We have developed a method to extract a hierarchy of clusters which is presented in Section 3.3. We extract region names by integrating Foursquare and Flickr data. In additional to individual places, such as buildings, monuments or other attractions, we are able to annotate larger areas like parks, neighborhoods or islands (Section 3.4). Finally, we describe how trajectories are annotated using the hierarchy of named regions of interest (ROIs) and evaluate their performance in this task in Section 3.5.

3.1 Place Semantics with Points of Interest

The most common method of stay point identification is to use a POI database, sometimes called a gazetteer, and select a POI that is within some distance r from the stay point, for example $r = 100m$. When there are multiple POI candidates, the closest one is chosen [96, 23, 12, 76, 69].

The main problem with this association method is that POI databases typically are not designed for tourist trajectory annotation. They may contain local businesses, sport clubs, academic buildings and other types of places, often with ambiguous and sparse metadata. It is therefore necessary, at minimum, to build a database specifically for annotation. If the place category metadata is fine-grained and accurate enough, POIs can be filtered to include only the ones relevant to the application. An example of such a database with rich category metadata is Foursquare. Querying Wikipedia for geo-referenced pages within a bounding box and clustering objects very close to each

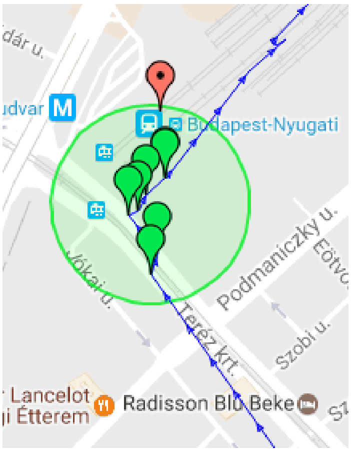

Figure 7: A cluster of photos (green) on a trajectory (blue) associated with a POI (red marker). Map data by Google.

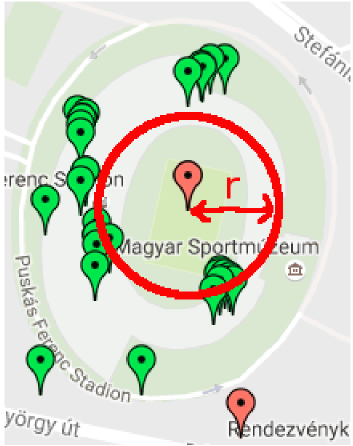

Figure 8: Associating photos (green markers) with places by proximity to coordinates (in red) disregards the actual extent and shape of the location. Map data by Google.

other to find POIs was proposed by Brilhante et al. [12]. In their definition, a POI can group nearby objects and therefore has some overlap with our concept of a ROI. For the Sightsmap.com application we integrated photo density information with other sources, for example Wikipedia where the editing policies are aimed toward including only generally notable places [114].

Even when the database of POIs is customized for semantic annotation, ambiguity is still possible when there are several candidate POIs nearby. We propose to differentiate between places by their importance in the application. For tourist recommendations, we measure the importance by popularity. Let $l \in L$ be a place in the set of places L within a given radius r. The distance between a place and the stay point is r_l and the popularity of the place is $popularity_l$. We choose the associated place by

$$\arg\max_{l \in L} \frac{popularity_l}{r_l^2} \tag{3}$$

The same formula but with the application of estimating place transition probability was simultaneously proposed by Maeda et al. [86]. In both our work [51] and by other authors [86, 35] this model of place "attraction" has been called the gravity model.

The stay point in our application of trajectory annotation is the centroid of the cluster of photos. Equation 3 tends to select the most popular place close to the centroid. Figure 7 illustrates the result of semantic annotation with POIs, where a cluster of photos (green empty markers) on a trajectory is associated with a nearby POI (red dotted marker).

The gravity model that we propose for POI based annotation was validated in a small scale experiment. We manually annotated 100 trajectories of Panoramio users in Budapest to use as the ground truth. We then annotated the trajectories with both the proximity and gravity based methods and compared the results to to the ground truth. Both the manual and automated annotations were done using the POI database of Sightsmap [114]. For the baseline proximity method, we set $r = max(100m, d_{max} + 10m)$ where d_{max} is the maximum distance between the cluster centroid and a cluster photo. For the proposed gravity method, we set $popularity_l = n_l + 1$ where n_l was the number of place visits in the ground truth data and $r = 2max(100m, d_{max} + 10m)$.

The methodology in this experiment was designed to automatically validate the results against the ground truth and has some important differences to the experiments presented in Section 3.5. The annotation accuracy in this experiment and the later ones cannot be compared directly.

To allow different clusterings, we compared annotations for each individual photo. The annotation was considered correct, if it matched the ground truth. However, we additionally considered the photos with no associated place correct if the ground truth also had no place associated. This method was chosen because we preferred that in areas where the Sightsmap POI database had poor coverage, no place visits would be inferred, instead of invalid place visits. In later experiments we abandoned this method in favor of total trajectory coverage.

Out of the total of 4360 photos, the baseline proximity method annotated 2595 (60%) photos correctly, while our proposed gravity method had 2981 (68%) correct annotations. Since this was a small scale experiment, we also calculated the p-value $p \ll 0.001$ [51, 33] confirming that the result was statistically significant.

With the popularity measure used as a weight, the gravity method obviously has popularity bias. While this is not a flaw, the question arises whether a performance metric that rewards popularity bias is appropriate in this experiment. The popularity bias is mainly a concern in the output of the recommender that the user is directly exposed to. The output of the annotation is not yet a recommendation, but is used as the input of the recommender for the purpose of modeling user behavior. If we consider a metric such as item coverage more important here, we emphasize correctly identifying the "long tail" but would not differentiate between missing a place with 1 visit from a place with 20 visits. The resulting statistical model could more easily misrepresent the place visiting distribution. Therefore the use of accuracy metric is appropriate here and the experiment confirms that the proposed gravity method outperforms the commonly used proximity association in tourist trajectories.

3.2 Region Discovery

The ambiguities in place association that the heuristics in Section 3.1 attempt to address are related to poor spatial description of places in POI databases. Real world geographical features have size and shape, as illustrated in Figure 8. The stands of a football stadium, where most photos would be taken from, form an "O" shape. In the illustration, with $r = 100m$ most of the photos would be too far from the central marker that belongs to the POI. Tuning the value of r inevitably means balancing trade offs, since increasing r would mean lowered accuracy with smaller spatial features due to more potentially irrelevant POIs being included.

In Sections 3.2–3.4 we introduce a new method we developed for semantic annotation. We represent places as geographical regions of interest (ROIs). The novelty of the method is the hierarchical, or layered structure of the ROIs which allows describing both small features which we call places, and larger features we call areas. Areas may contain multiple places. Our proposed method that includes region extraction, naming and trajectory annotation, is called the hierarchical ROI (HROI) method. The method also includes an algorithm that extends the HDBSCAN density-based clustering to extract hierarchical layers of clusters. We evaluate our proposed HROI method, compared to POI based annotation, in Section 3.5.

The hierarchical regions of interest (ROIs) are extracted from geo-tagged photos as clusters. This identifies areas where photos are taken more frequently. The density of the photos can be thought of as a proxy indicator of the popularity of any given area [26].

There are both general-purpose (DBSCAN[32], OPTICS[7]) and specialized algorithms (P-DBSCAN[57]) that can be used to create a single clustering of photos. The main drawback of general purpose methods is that they are either unable to cope with local density variation (DBSCAN) or require manual intervention to produce the best clustering (OPTICS). This problem is addressed in P-DBSCAN that automatically adapts to local changes of density. P-DBSCAN still cannot cope with extreme variation of density, but in our experiments produced competitive results [53].

Using any algorithm that creates a single clustering requires parameter tuning to produce a multi-layered hierarchy of clusters such that each layer contains clusters of appropriate size. We therefore approach the problem by using hierarchical clustering, where the formation of multiple layers is data driven. We use the HDBSCAN algorithm to create a tree of clusters and develop a new algorithm called HDBSCAN/n to compress the tree to a given number of layers such that the hierarchy is preserved. The details of the algorithm are given in Section 3.3.

The distance between geographical coordinates does not always represent accurately how related two points are to each other in segmented urban spaces. For example, wide motorways that are present in most cities form obstacles for pedestrians. In historic cities, places that are separated by walls or fortifications may not be well connected. Wider rivers separate regions that may have different historical demographics, were developed in different eras and therefore have different appeal and features.

The points that are being clustered are photos that have associated metadata. The photo tags can indicate, whether the two photos belong to the same physical place or area. Finally, because our experiments are being conducted using Flickr data, the photos also have Flickr's "Where on Earth" (WoE) identifier that is associated with the name of the neighborhood or part of the city.

To include additional geographical (natural and artificial obstacles) and semantic (photo tags, WoE id) information we introduced a linear combination distance function between points x_i and x_j:

$$d_s(x_i,x_j) = d_{osrm}(x_i,x_j) + \beta d_{WoE}(x_i,x_j) + \gamma d_J(x_i,x_j) \tag{4}$$

The parameters α, β and γ are chosen empirically. d_{osrm} is the routing distance between the two points. We use OpenStreetMap and Open Source Routing Machine (OSRM) [85] to calculate pedestrian walking distance. This makes it more difficult for points separated by obstacles to be included in the same cluster.

d_{WoE} in our proof of concept implementation is 0 when the points have the same "Where on Earth" identifier and 1 otherwise.

The semantic similarity between two points is calculated by the Jaccard distance between the sets of tags T_i, T_j of the two photos:

$$d_J = 1 - \frac{|T_i \cap T_j|}{|T_i \cup T_j|} \tag{5}$$

Our proposed distance function with semantic components (Equation 4) has side effects that would not be a concern with a simple spatial distance function. The function does not have the properties of a true metric, most notably it does not have the triangle

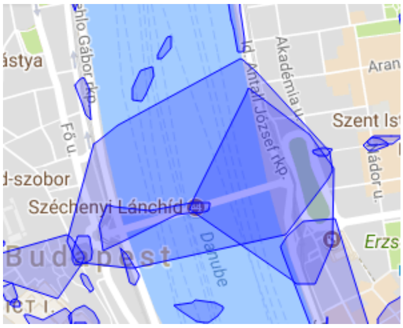

Figure 9: Spatially overlapping regions, caused by clustering with a distance function that includes semantic distance. Two points at the same geographical location may appear as apart from each other to the clustering algorithm. Map data by Google.

inequality property. This complicates spatial indexing, which makes clustering slower. We have addressed the technical aspects of implementing the clustering efficiently in [53].

The second side effect is that clusters that do not have a parent-child relationship may overlap in map coordinate space (Figure 9). The effect is undesirable, because it complicates semantic annotation. Having several unrelated clusters covering a point makes the choice of which ROI the point belongs to ambiguous.

3.3 Hierarchical Clustering

In this section we give an overview of the HDBSCAN density-based clustering algorithm and then describe the extension to the algorithm that we developed for extracting a layered hierarchy of clusters. We use the new extended algorithm together with the semantic distance function we introduced (Equation 4) to find the hierarchy of frequently visited spatial regions.

The HDBSCAN Algorithm

In density-based clustering, the data points can be thought of as leaf nodes in a tree. Interior nodes represent the clusters that include all the leaves under the interior node. Each interior node corresponds to a specific density threshold where all the leaves under it are close enough to be within the threshold according to some definition of density. The cross-section of the tree at some density threshold represents a clustering of the data points (Figure 10, a.).

Some algorithms, like DBSCAN, define the density threshold but do not explicitly create the tree or even require the concept [32]. Others, like OPTICS, create the tree and leave the task of choosing the appropriate threshold to the user [7]. The innovation of HDBSCAN by Campello, et al. was the way to automatically find the optimal clustering by introducing the concept of cluster stability [15].

Campello et al. proposed that we choose the clusters in such a way that the sets of points belonging in a cluster remain relatively unchanged across as wide a range of density as possible. They measure this by stability. As the density required to form a cluster is decreased, clusters appear and are gradually joined together. The lifetime of the cluster C_i over the continuously decreasing range of density is defined by the two parameters of HDBSCAN. The cluster is created when it contains at least m_{pts} points or when two clusters are joined together each containing at least m_{clSize} points. The cluster ceases to exist at the density threshold where it is joined into a new cluster.

Let $x_j \in C_i$ be a point in a cluster C_i. The density level function $\lambda_{max}(x_j, C_i)$ is the maximum density at which x_j still belongs to C_i. The function $\lambda_{min}(C_i)$ is the minimum density level where C_i exists as an independent cluster. The cluster stability is then [15]:

$$w(C_i) = \sum_{x_i \in C_i} \Big(\lambda_{max}(x_j, C_i) - \lambda_{min}(C_i)\Big) \tag{6}$$

In HDBSCAN, the cluster is defined as a maximal set of *density-connected* points. These are points that have at least m_{pts} neighbors closer than the core distance ϵ and each pair of points is either at most the distance ϵ from each other, or connected by a path through the cluster members where each step is less than ϵ. The density level in Equation 6 is defined through the core distance by $\lambda = \frac{1}{\epsilon}$.

HDBSCAN creates the optimal clustering C^* that maximizes $\sum_{C_i \in C^*} w(C_i)$ such that for $\forall C_i, C_j \in C^*$ the cluster C_i is not on the path from C_j to the root of the tree (Figure 10, b.). The latter ensures that each point belongs to a single cluster, but does not require that the clusters have the same density. This property is important for finding ROIs, where local density can vary considerably between busy tourist hot spots and larger recreational areas.

Our Extension to the HDBSCAN Algorithm

Briefly, HDBSCAN forms a single layer of clusters that persist over a range of densities and allow local density variations. This can be used to extract a single layer of ROIs by using only a single parameter $m_{pts} = m_{clSize}$. For the concept of the hierarchical representation of the city space, we require the creation of multiple layers. To take advantage of the automated density selection through cluster stability and the support for local density variation, we developed an extension to HDBSCAN to produce a n-layered clustering, called HDBSCAN/n. It is a general purpose algorithm and has applications going beyond the scope of this thesis.

We define the n-layered optimal clustering as the set of clusters C_n^* such that $\sum_{C_i \in C_n^*} w(C_i)$ is maximized and for all $C_i \in C_n^*$ there are no more than n clusters on the path from C_i to the root of the cluster tree (C_i included). Because the following discussion uses the concept of tree depth, we define the depth of the root node as 0 and the depth of the tree d as the depth of its deepest node.

When $d < n$, creating the clustering is trivial, because all of the clusters need to be included. For example, to make a two-layered clustering for a tree with $d = 1$, both the root node and its children have to be included to maximize the sum of $w(C_i)$. Assuming $n = 1, 2, \ldots$ any tree will contain subtrees with $d < n$ and for such subtrees the optimal clustering is the complete subtree.

In Figure 10, c., the subtree $\{C_4, C_6, C_7\}$ is trivially solved for two-layered clustering. An optimal single layer clustering would be the set $\{C_6, C_7\}$, because these clusters can be selected together. If C_4 is selected, C_6 and C_7 have to be discarded, because C_4 shares a path with both of them. The stability of C_4 is lower than the combined stability of C_6 and C_7.

Observing that subsets of clusters are mutually exclusive due to being on the same paths, we designate $S_1 = \{C_6, C_7\}$ and $S_2 = \{C_4\}$. The solution $C_1^* = S_1$ and the set S_2 is the clusters we can add if another layer is allowed, so $C_2^* = S_1 \cup S_2$.

For disjoint trees, the optimal solution is the union of the optimal solutions of individual trees. Any other solution would necessarily have lower total stability. Therefore,

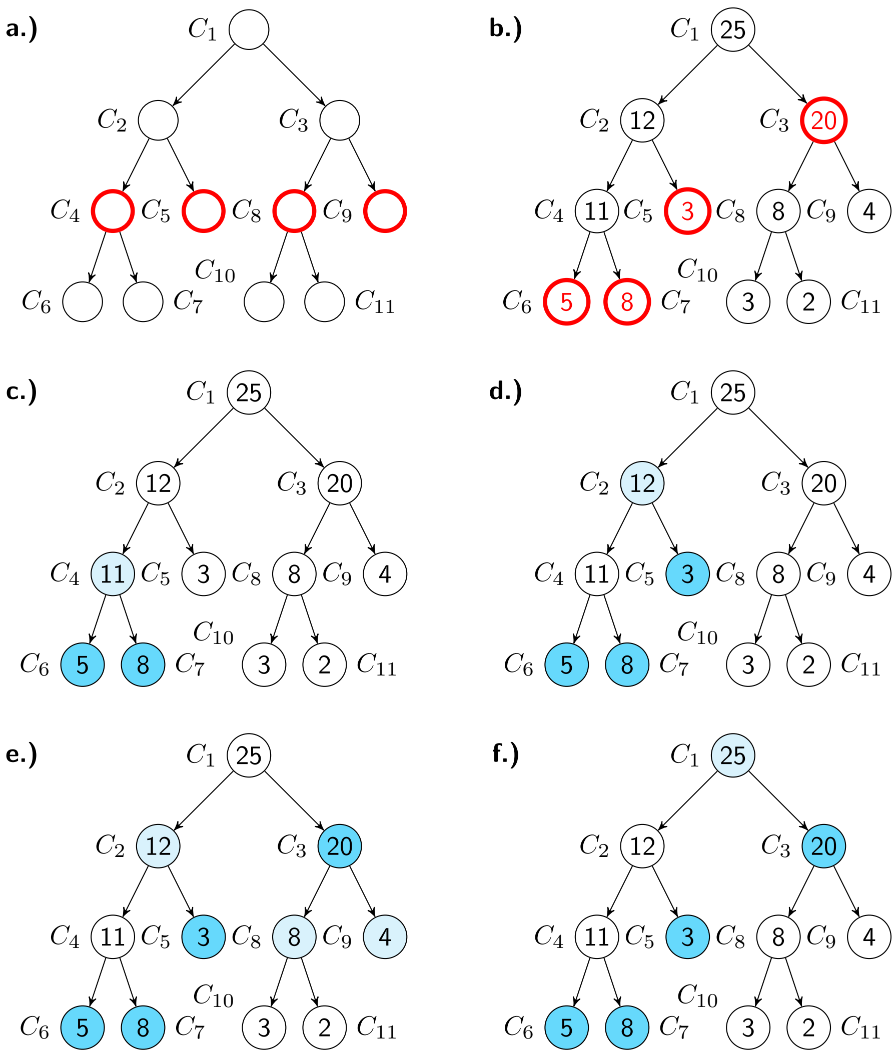

Figure 10: Hierarchical density based clustering. For simplicity, clusters at the same depth are assumed to have the same density. The nodes with thick outline indicate the chosen clustering. a.) *Clustering by a cross-section at a density threshold.* b.) *Clustering with HDBSCAN that maximizes $\sum w(C_i)$. The stability $w(C_i)$ is the number inside the nodes. The graphs c.–f. show the steps of forming the cluster hierarchy with our proposed extension to HDBSCAN.* c.) *Beginning of the solution to maximize $\sum w(C_i)$ when two layers are allowed. The subtree $\{C_4, C_6, C_7\}$ is solved. Deeper fill is the set S_1 and lighter fill is S_2. $S_1 \cup S_2$ is the solution for the completed subtrees at any given time.* d.) *Solution after joining subtree $\{C_5\}$ at the node C_2.* e.) *Both subtrees of the root have been solved from leaves up.* f.) *Final solution after joining the subtrees at the root C_1. Set S_1 is the same as the HDBSCAN solution.*

we can simply grow our intermediate solution over already-solved trees with no connecting parent. In Figure 10 the subtree $\{C_5\}$ also has a trivial solution, so $S_1 = S_1 \cup \{C_5\}$ and $S_2 = S_2 \cup \emptyset$ for the children of C_2.

However, when we include the cluster C_2 and look at the entire subtree starting from that node, some paths from the leaves to the root have more than two nodes, so some clusters cannot be part of the optimal solution. It turns out that in this case it is not necessary to examine all possible combinations of clusters as solution candidates. Let $S' = \{C_2\}$ be the new set of clusters including only the root node of the subtree we have examined so far. S_1 will always be part of the solution, because we know that if we need to discard clusters to reduce the number of nodes on each path, we should discard those in S_2 first. However, if the total stability of S' is lower than that of S_2, we should keep the current solution.

In Figure 10, d. the subtree starting from C_2 is solved, as the new parent node replaces the old S_2. To solve the entire tree, we traverse it in depth first order. Each time we go up a level, we first join the solutions from sibling subtrees (Figure 10, e.) and then make a decision about the parent (or root) node. Figure 10, f. gives the complete solution $C_2^* = S_1 \cup S_2$ where S_1 is the same as the solution for HDBSCAN.

Algorithm 1 gives the general procedure that we developed for creating n-layered clusterings by keeping track of sets $S_1 \ldots S_n$ as a priority queue. Each time we encounter a new parent node that joins already solved subtrees, we compare the stability of the set containing the new parent S' to each of $S_1 \ldots S_n$. If it exceeds the total stability of any of the existing sets, it will be placed into the priority queue at that position, and the currently last set S_n will be discarded. While the problem of maximizing the stability is combinatorial, by using the priority queue we can solve it in a single depth first traversal, with time complexity $O(N)$ where N is the number of clusters. The correctness proof for this algorithm has been sketched in our conference paper [53].

Algorithm 1 The proposed extension of the HDBSCAN density based clustering algorithm to form a n layer clustering (HDBSCAN/n).

function $\text{MAX\_WEIGHT\_SUBSET}(n, C_{root})$
    Initialize $S_j \leftarrow \emptyset$, $j \in 1, \ldots, n$
    for each child C_i of C_{root} **do**
        $S_1^i \ldots S_n^i \leftarrow \text{MAX\_WEIGHT\_SUBSET}(n, C_i)$
        for $j \leftarrow 1, n$ **do**
            $S_j \leftarrow S_j \cup S_j^i$
        end for
    end for
    Find smallest k such that $w(C_{root}) > \sum_{C \in S_k} w(C)$
    $S_{j+1} = S_j$ for $j \geq k$
    $S_k = \{C_{root}\}$
    Return $S_1 \ldots S_n$
end function
$S_1 \ldots S_n \leftarrow \text{MAX\_WEIGHT\_SUBSET}(n,$ root of cluster tree$)$
$C_n^* \leftarrow \bigcup^n S_j$

3.4 Region Naming

Regions of interest represent hot spots that people visit and photograph, as well as larger areas that have sufficiently high density of photos. To make use of them in semantic annotation, we need to match each region to an appropriate real world object. In other words, we need to name the regions.

In this section, we describe the region naming stage of our proposed HROI method in detail. We integrate two types of information for region naming – known POIs from a database and the textual information the users uploading the photos have provided. The process of naming consists of the following stages:

1. Name candidate extraction. We find the names of Foursquare POIs nearby and also generate possible names from the titles and tags of the photos within the regions, by selecting n-grams of words used by the greatest number of distinct users.

2. Vectorized representation of regions. We represent each region as a vector of tag counts. We then use the latent semantic indexing (LSI) [29] transformation to create a matrix of region semantic vectors with reduced dimensions.

3. We find name candidates that are semantically similar to regions. Each region will be assigned the most similar name automatically, but we also use ad hoc heuristics to detect ambiguous cases and flag these regions. We record the confidence (semantic similarity) we have that the name is correct.

4. Human assisted resolution of a subset of cases. We use an application that displays the region, selected name and other similar names for heuristically chosen problematic cases, allowing the human expert to verify and correct the name assignments.

The similarity of name candidates and regions is done by comparing them in reduced dimensionality latent space. The idea behind using latent semantic indexing is that it will combine similar tags together into one dimension. By combining semantically similar tags, or terms, we allow themes, or topics of the regions and names to emerge automatically. This allows us to determine if the name is connected to similar topics as the tags combined from the pictures of the region.

We also use popularity weighting to prefer names that were used by more users. Denoting the number of users whose pictures contributed to the name (i.e. the name appears in their titles or tags) $N_{candidate}$ and the total number of distinct users in the region N, we calculated the weighted similarity as

$$s = (1-\alpha)s_{candidate} + \alpha s_{candidate}\frac{\ln N_{candidate}}{\ln N} \quad (7)$$

where $s_{candidate}$ is the cosine similarity between the name vector and the region vector. In case of POI names, the user counts are taken from check-in counts, so the weighted similarity prefers more popular POIs. The parameter α ensures that only names that are already similar to the region tags are considered, popularity weighting helps to decide between them. We empirically chose $\alpha = 0.5$.

There are several ways of detecting whether a name assignment based on semantic similarity could be problematic. In case of POI names, we can check if the POI is within the region, near it or very far. Many similar candidates with no clear winner

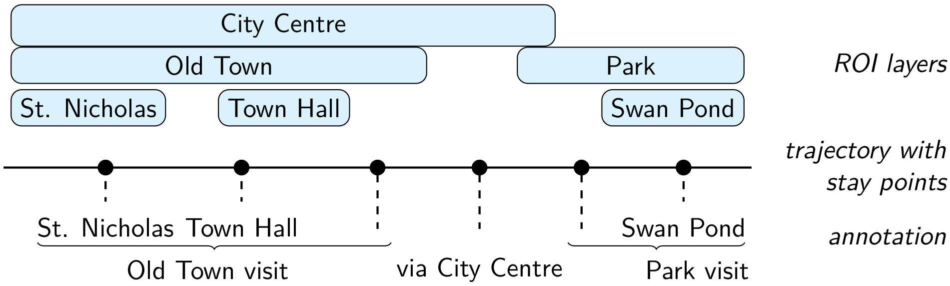

Figure 11: Trajectory annotation using regions of interest (ROI) layers. The higher layers allow annotation of the middle trajectory segment and provide context to individual place visits.

indicates that the program had to make a guess. Low similarity between the best name candidate and the region tags also suggests that the chosen name could be irrelevant.

Based on the above heuristics, we choose a subset of ROIs to be validated by a human expert. Additionally, we select regions that are large and dense, because these regions often contain one or several individual places that are frequently photographed. There is a tendency that larger regions are automatically named after these smaller scale places because of the high proportion of the tags related to then in the photos taken within the region. In our experiments we chose 90 regions for the human-assisted name resolution.

The human expert reviews the region boundaries and automatically assigned names. For each of the reviewed regions, they may select another name from the list of name candidates ranked by similarity, manually type the name or choose to keep the automatically assigned name. Using this process, the expert was able to decide the names of 90 regions in approximately 2 hours.

While human assisted annotation may not fit well within the paradigm of automatic knowledge discovery from big data, it is a pragmatic solution to a practical problem and similar approaches have been described in the literature [6]. In this process, the automatic and human resolution complement each other. The human expert contributes the most important knowledge to the process, while the mechanical processing of large amounts of photos and trajectories is left to the software. Two main problems remain, however. The automatically discovered boundaries of regions may not give the best results for semantic annotation. In our application, the expert can not adjust the boundaries. The method also does not scale beyond individual cities, and local expert knowledge is required.

3.5 Semantic Annotation with Regions of Interest

Named ROIs provide a straightforward way of semantic annotation. For each stay point on a trajectory, we find the region it lies within (Figure 11). The name of that ROI is then assigned to the point. In case there are multiple overlapping regions, we use the following tiebreak criteria:

- For a pair of regions, the one on the lower layer of the hierarchy is preferred, as it is expected to be a more accurate, finer-scale object.

- For a pair of regions that reside on the same layer, we compare the distance of the point to the centroid of the cluster forming the region and the confidence of the assigned name (semantic similarity of the name to the region).

We evaluated the proposed HROI method of semantic annotation by an experiment designed to answer two research questions:

1. Does the annotation with ROIs perform better than heuristic matching against POI databases commonly used in published studies?

2. Does the HDBSCAN/n algorithm, designed to produce multi-layer hierarchies, perform better than simply stacking layers created with an existing clustering method?

The task in the experiment was to annotate trajectories extracted from Panoramio photos in four European cities: Budapest, Tallinn, Venice and Vienna. For ROI shape discovery we used geo-tagged photos from Flickr and for ROI naming Flickr metadata and Foursquare places.

For comparison, we included three POI-matching baselines. The first was to query Google Places for nearby POIs and choose the closest one (*GP proximity*). The second method also used Google Places, but used Google's opaque "most prominent" sorting method to find the most relevant POI nearby (*GP prominence*). In both cases, we attempted to use Google's POI categories to filter out irrelevant places. The third POI based method (*4sq gravity*) used Foursquare places that we associated with the "gravity" method described in Section 3.1. Foursquare has a fine-grained category system that we used in filtering out places that are typically not tourist attractions.

To evaluate region shape discovery, we added the P-DBSCAN algorithm as another baseline. We formed a 3-layered clustering using P-DBSCAN to compare HROI against an alternative method of ROI shape formation. P-DBSCAN is designed for region discovery from geo-tagged photos and has an adaptive mechanism for coping with differences in local density [57]. For each layer, we empirically found parameters that formed place, larger feature (park, beach) and neighborhood level features [53]. Region naming was done according to Section 3.4.

The summary results of the experiment are shown in Figure 12. HROI is short for the proposed hierarchical ROI method with the HDBSCAN/n clustering.

We measure accuracy on two levels of granularity. If the place was correctly identified, then we consider the annotation correct on place level. Since the hierarchical approach is designed to explain parts of trajectories that may lie in areas between hotspots, we separately count annotations that are correct on area level. Some POI based annotations also received names which represented larger objects so we counted the correctness of those on area level.

The ability to annotate on area level gives a clear advantage to hierarchy-based methods in terms of overall points annotated correctly. This satisfies one of the initial goals of designing the ROI method, namely to increase the coverage of annotations. However, there was no overall advantage in place level accuracy. The Foursquare gravity POI association by gravity performed slightly better (49%) than the proposed HROI method (46%) in assigning place level annotations.

Comparing the results of HROI and P-DBSCAN, the trade off between place level and area level accuracy is apparent. With P-DBSCAN, hand-tuning the parameters by visually examining the cluster boundaries appears to have been unsuccessful in terms of discovering place level structure, which highlights the advantage of the data driven

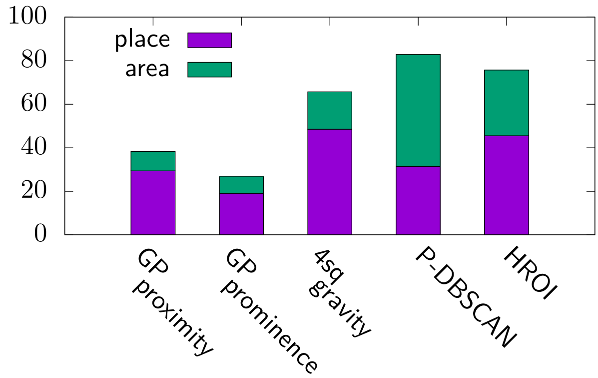

Figure 12: Overall annotation accuracy. The bars represent the percentage of places annotated accurately at place level and at area level. Our proposed HROI method provides the best compromise between covering a higher ratio of stay points and maintaining place level accuracy.

approach of HROI. At the same time, annotations on area level were the most accurate overall.

For POI based methods, the source of the POIs plays a major role in annotation accuracy. For the cities that were part of the experiment, the quality of place metadata in the Google Places API was so poor that we could not perform accurate filtering. In many cases the stay points were annotated with the names of locally registered small businesses which are irrelevant to the tourist.

The accuracy results give us an indirect measure of the quality of region shape formation. We also designed another experiment where over 300 survey participants were asked to rate the quality of region shapes on the map in terms of how well they represent known objects. The survey established that the participants rated the regions created by hierarchical approaches more highly, preferred in 51% of the cases where a hierarchical and non-hierarchical clustering were shown side by side. Flat, or non-hierarchical clustering was rated better in 36% of the cases. The survey showed no significant differences between the clustering algorithms [53].

In conclusion, hierarchical ROIs give better trajectory coverage than using public POI databases. We observed a small decrease in place level accuracy, but its effect depends on the application. If it is acceptable in the application to recommend more general areas, such as neighborhoods, historical parts of the cities or recreational areas, the loss of place level accuracy is not important. The hierarchical representation of the urban space resulted in more accurate region boundaries than a single layer of ROIs, according to the region shape evaluation survey.

In terms of clustering methods, the proposed HDBSCAN/n outperformed P-DBSCAN in place level annotation accuracy. This is explained by the practical difficulty of tuning the P-DBSCAN clustering for the annotation task. Therefore, the advantage of HDBSCAN/n appears to be mainly in automation of the creation of multiple layers. In area level accuracy and region discovery, HDBSCAN/n did not perform better.

4 Evaluation of Impact on Recommendations

In Chapters 2 and 3 we have introduced methods of noise filtering and semantic annotations of tourist mobility data extracted from geo-tagged photos. In these chapters we evaluated how well these methods performed in their designated task. Ultimately, the purpose of developing these methods was to improve the recommendations. In this chapter we evaluate the impact the filtering and annotation methods have on the recommendation.

We experiment with a set of recommenders that have been widely used in applications and thoroughly described in the literature. In the evaluation, we are interested in the changes the training data preparation cause for each individual recommender. Hence, in the evaluations it is not important which recommender model performs the best, but rather, which training dataset causes the recommenders to perform the best.

In Section 4.1 we describe the sequential and session-based scenario of recommendation that we have assumed. We use recommendation models that include both well-known baselines and methods that have been shown to have high performance in the session-based scenario. The experiments use offline evaluation metrics. We define the metrics as well as explain their place in wider context of recommender evaluation in Section 4.2. In Section 4.3 we introduce predictability, a new metric specifically designed for sequential datasets. The results of the evaluation are given in Section 4.4.

4.1 Making Recommendations

The task of place recommendation is sequential in nature. Many tourist recommendation systems [23, 61, 76, 12, 69, 18, 21] and models of recommendation [122, 37, 42] focus on building trip itineraries. In a simpler but still relevant case of next-place recommendation, we typically assume that there is a previous history of places that the recommendation can be based on.

We have assumed a session-based scenario where no information is known about the user. There are two reasons for this - first, we consider the session-based approach of practical value because it does not require the system to collect and store personally identifiable data. Second, in our city-scale evaluation datasets, tourists rarely return for another visit and therefore including the user in the recommendation model adds little value. We treat trajectories annotated with place visits as sessions and individual stay points as session items.

Predictive Models

The recommenders we use are designed to build a predictive model using a set of training sessions in order to predict the next item in an incomplete test sessions.

Markov chain(MC) learns the transition probabilities between session items from observations in the training data. Denoting the set of sessions where location l_j follows l_i in a sequence with S_{l_i,l_j} and the set of locations with L,

$$P(l_j|l_i) = \frac{|S_{l_i,l_j}|}{\sum_{k \in L} |S_{l_i,k}|} \tag{8}$$

gives the probability of transition from l_i to l_j. The ranked list of recommendations is the list of all transitions with a non-zero probability, in descending order of probability.

Unless otherwise noted, we use a first order Markov chain defined above. A second order Markov chain is defined similarly by

$$P(l_k|l_i,l_j) = \frac{|S_{l_i,l_j,l_k}|}{\sum_{k \in L} |S_{l_i,l_j,k}|} \tag{9}$$

where S_{l_i,l_j,l_k} is the set of sessions where l_k was visited after the sequence l_i, l_j. In case the current session is shorter or the preceding sequence was not present in the training set, the recommender is allowed to fall back to 1st order Markov chain.

Association rules(AR) similarly learns transition probabilities from co-occurrences in the training data, but does not consider the sequence of items. The more often an item has been in the same session with another, the more likely is the transition to that item. This is useful when we do not wish to emphasize the order of user actions. For location l_i, we give the recommendations in descending order of the ranking score r_k for items $k \in L$. For a particular transition l_j, the ranking score is

$$r_{l_j} = |S_{l_i} \cap S_{l_j}| \tag{10}$$

where S_k is the set of sessions where an item $k \in L$ occurred. The transition probability is

$$P(l_j|l_i) = \frac{r_{l_j}}{\sum_{k \in L} r_k} \tag{11}$$

Session-based k-nearest neighbors(SKNN) is a neighborhood-based method that, unlike the above baselines, addresses the sparsity in training data. Possible transitions include items that not only occurred together with the current item, but those that occurred in sessions similar to the current session. Denoting the current session s and the set of nearest neighbors N, the ranking score is computed as

$$r_{l_j} = \sum_{n \in N} sim(n,s) 1_n(l_j) \tag{12}$$

where $sim(n,s)$ is the binary cosine similarity calculated by representing sessions as vectors in item space. The $1_n(k)$ is the function indicating the membership of item k in the session n. The set of nearest neighbors consists of sessions most similar to s, determined using the binary cosine similarity function. [11]

Vector Multiplication SKNN(VSKNN) is a sequence-aware modification of SKNN that places more importance on more recent items in the current session [83]. The similarity function is modified to include a decaying weight for items earlier in the session:

$$sim(n,s) = \frac{\sum_i n_i s_i \frac{pos(i)}{|s|}}{\sqrt{\sum n_i^2} \sqrt{\sum (s_i \frac{pos(i)}{|s|})^2}} \tag{13}$$

Where n_i and s_i are elements of the vector representation of sessions n and s and $pos(i)$ is the position of the item corresponding to index i in the current session s, or 0 if not in the session. The ranking function is also modified [83]:

$$r_{l_j} = \sum_{n \in N} sim(n,s) \frac{1}{|s| - pos(l_j) + 1} 1_n(l_j) \tag{14}$$

The MC and AR methods are included as baselines that represent the opposites in terms of emphasis on sequence. The SKNN and VSKNN methods represent the state of the art performance in session-based recommendation [83, 84], at the same time remaining simple and transparent models that are space and time efficient.

4.2 Performance Metrics

Evaluation of recommenders is usually done using offline, user study or online methods. Offline methods use benchmark datasets to measure recommender performance in terms of accuracy of predicting user actions, but also metrics such as diversity and coverage. User studies collect feedback from a test group of users of the recommender. The online methods collect metrics from a deployed recommender application.

Offline methods, especially those that focus on prediction accuracy have been recognized to have limitations [94, 113]. Their widespread use is mostly due to being the least expensive method to do extensive evaluations. In some cases, the accuracy of predicting user actions has also correlated strongly with direct user feedback [27]. Offline methods also have better reproducibility when following good research practices [28].

We use offline accuracy metrics in our impact evaluation. The results should be viewed and interpreted with the consideration that they reflect how predictable are user actions using our metrics. The utility in an actual recommender application may involve other factors. For instance, the evaluation does not reflect how well the semantic annotation copes with practical issues such as place name deduplication and language [77, 78].

In our experimental setup the recommender is given the task to predict the next item in an incomplete sequence of visited places. The recommenders produce a ranked list of items P with the most likely item in the first position. The hit rate (HR@n) metric measures the ratio of tests where the correct item appeared in the top n of the ranked list:

$$HR@n = \frac{1}{|S|}\sum 1_{P_n}(l) \tag{15}$$

Where S is the set of test cases and $1_{P_n}(l)$ a function that indicates whether the correct item l is in the top-n set of predictions P_n for each test case. Since the HR@n metric does not differentiate between item positions inside the top-n, in more recent experiments we opted for the mean reciprocal rank (MRR@n):

$$MRR@n = \frac{1}{|S|}\sum 1_{P_n}(l)\frac{1}{r_l} \tag{16}$$

Where r_l is the rank of l in the ranked list P.

4.3 Predictability Metric for Datasets

The datasets used in sequential recommendation have regularities, introduced for example by human habits, personal preferences and trends. Despite these regularities there are always unknowns. We do not know the true causes behind user decisions; additionally in session-based recommendation the users are anonymous. Such unknowns manifest themselves as uncertainty, meaning that from the point of view of recommender systems user actions are a stochastic process.

The prediction accuracy of recommenders varies greatly between different datasets. The question is then, how much the dataset itself contributes to the prediction error. We introduce the metric of dataset predictability for the sequential recommendation task that numerically measures the randomness in datasets.

We define *predictability* as the probability that the recommender will correctly predict the next item, given an unfinished session and a history of other sessions. The probability can be understood as the average ratio of tests where the recommender succeeds in delivering the correct prediction in relation to some metric, for example HR@n. In the simplest case, the prediction is correct if the recommender places the correct item first in the ranked output list.

Song et al. developed a method of finding maximum predictability of human mobility traces [112]. They derived Equation 17 that gives the relation between the entropy rate $\mathcal{H}(\mathcal{X})$ of a stochastic process and the maximum predictability of the sequences produced by the process.

$$\mathcal{H}(\mathcal{X}) = -\Pi^{\overline{max}} \log_2 \Pi^{\overline{max}} - (1 - \Pi^{\overline{max}}) \log_2 \frac{1 - \Pi^{\overline{max}}}{m-1} \tag{17}$$

where m is the number of unique items in the sequences. A thorough explanation of the derivation and an example of refining the predictability limit by adding contextual information was given by Smith et al. [110]

Equation 17 can be applied to find the upper bound on predictability for recommender system datasets that contain many sequences, by making simplifying assumptions, a.) that the sessions in the datasets are generated by the same stochastic process; and b.) even if the sessions sometimes occur in parallel in the real world, we treat them as if they had occurred strictly sequentially, i.e. we define an ordering on the parallel sessions based on the time stamps of items in sessions.

For practical purposes, the theoretical $\mathcal{H}(\mathcal{X})$ is replaced with an estimate of entropy rate [128]:

$$S = \left(\frac{1}{n} \sum_i \Lambda_i \right)^{-1} \log_2 n \tag{18}$$

Where $S \approx \mathcal{H}(\mathcal{X})$ is the estimate over a sequence of length n. $\Lambda_i = k_{max}^{(i)} + 1$ and $k_{max}^{(i)}$ is defined as the length of longest sub-sequence starting from position i that appears as a continuous sub-sequence between positions $1 \dots i-1$. The quantity S can be plugged into Equation 17 which can then be solved for the estimate of the upper bound on predictability, $\Pi^{\overline{max}}$.

The estimated predictability is a limit on recommender performance measured by the HR@1 metric. It applies to any algorithm, however individual algorithms may have different and lower limits, caused by factors such as the lack of sufficient training samples required by the algorithm [52]. Since the aim in this chapter is to compare dataset preparation methods, we only use the predictability limit determined by randomness as an algorithm-independent measure.

4.4 Experiments

The experiments were designed to measure the impact of filtering and annotation techniques on the performance of the recommenders. We assume a session-based recommendation scenario, where the recommenders learn from a set of anonymous annotated trajectories. The recommender is then tasked with predicting next place visits in incomplete trajectories.

We begin with unannotated trajectories extracted from geo-tagged photos in different cities. We then prepare training sets using different combinations of the evaluated methods and measure recommender accuracy using each training set. In general recommender system terminology, trajectories are sessions and place visits correspond to items.

The experiments should confirm three hypotheses: 1.) Filtering improves recommender accuracy due to removing noise in the training data; 2.) Semantic annotation using our proposed popularity heuristics improves recommender accuracy by differentiating between actual tourist destinations and irrelevant places; 3.) The semantic annotation method with hierarchical regions that we developed improves recommender accuracy by covering more of the trajectories and avoiding unnecessarily high annotation granularity in places where interesting objects are densely located.

Datasets

We used the following datasets as the source of tourist trajectories for annotating. The visit sequences in annotated trajectories (except the testing subset) were used to train predictive models.

- Panoramio photos from Budapest (October 2002–July 2016, 4061 sessions with more than one photo).
- Flickr photos from Budapest (January 2003–November 2017, 10277 sessions).
- Flickr photos from Tallinn (March 2006–September 2017, 2883 sessions).

As sources for information about POIs and regions, we used the following data:

- The dataset of Sightsmap [114] that includes coordinates and popularities of points of interest (POIs) worldwide. The data is derived from Panoramio, Foursquare and Wikipedia.
- POIs from the Foursquare API. The popularity is estimated from the number of check-ins.
- ROI shapes and names generated from Flickr and Foursquare data using the Hierarchical ROI method, as described in Sections 3.2–3.4.

Evaluation of the Gravity Model

In this experiment, we compared semantic annotation with the widely used proximity heuristic (closest POI in some radius r) [96, 23, 12, 76, 69] and the gravity heuristic we proposed [51]. The gravity heuristic is described in Section 3.1 and uses POI popularities as weights in annotation candidate selection. We also measured the effect of noise filtering heuristics as introduced in Section 2.2.

In the experiment we prepared four training datasets, combining filtered and unfiltered traces of Panoramio users in Budapest with two annotation heuristics. The POI data from Sightsmap was used for annotation. Additional 100 sessions were manually annotated to form the test dataset of sessions.

We let a 2-nd order Markov chain recommender predict the next item after each item in the test sessions iteratively. Total number of transitions to predict in the experiment was 805. In our earlier publication [51] we only reported results with unfiltered trajectories using the HR@3 metric. Table 3 gives the complete results of the experiment. The gravity heuristic outperforms the proximity heuristic. This indicates that including popularity information allows more accurate annotation, as the model was better able to anticipate the transitions in manually annotated trajectories. There is no obvious positive or negative effect from noise filtering.

Table 3: Prediction accuracy on tourist trajectories in Budapest, from Panoramio.

Filtered	*Annotation*	*HR@3*	*HR@1*
no	gravity	0.371	0.222
yes	gravity	0.365	0.227
no	proximity	0.250	0.148
yes	proximity	0.237	0.157

Evaluation of Annotations with Hierarchical ROIs

To test the hypothesis that using ROIs improves the quality of annotated trajectories as the input data for recommender training, as well as obtain more conclusive results about the noise filtering in trajectories, we measured the impact of these preparation methods on model prediction accuracy. We experimented on three datasets: Flickr photos from Budapest and Tallinn, and Panoramio photos from Budapest.

For each dataset, we split the data to training and test splits. The training splits were prepared using four different methods, while there was only one version of the test split in each case. We trained well-known sequence prediction models (MC, AR, SKNN, VSKNN) using each differently prepared training set and evaluated them on the testing set. If our proposed HROI method produced higher quality training data than the POI annotation method, then the predictive accuracy of the models should increase when trained with the trajectories annotated with HROI. Similarly, if noise filtering improves training data, models trained on filtering data should have higher prediction accuracy. We also directly the measured predictability of the training/test splits.

The selection of training and test trajectories was done temporally. We divided each set of trajectories by time stamps into five roughly three-year training windows, interleaved with 30-day testing windows. This emulates the deployment scenario where the recommenders are trained on past sessions and give recommendations for current sessions. The five-way split reduces selection bias.

The four methods to prepare the training data were as follows:

- Unfiltered and annotated with the gravity heuristic and Foursquare POIs (abbreviated as *raw 4sq*).
- Unfiltered and annotated using the region data derived with the HROI method (*raw HROI*).
- Filtered with $v_{cut} = 10km/h$ and $r_{noise} = 0.15$, as described in Section 2.2 and annotated with the gravity heuristic and Foursquare POIs (*filt 4sq*).
- Filtered as above and annotated using the region data derived with the HROI method (*filt HROI*).

To create ground truth data, we manually annotated the test set of trajectories and converted them to testing sessions. The annotation was done in parallel using both Foursquare POIs and ROIs. Because neither method could cover each place visit correctly, we only accepted sessions where there was at most one consecutive place that did not have a correct annotation using either method, and no places that were left completely unidentified. Since that would have resulted in discarding a number of long trajectories, we allowed splitting trajectories into multiple sessions at points that could not be correctly annotated.

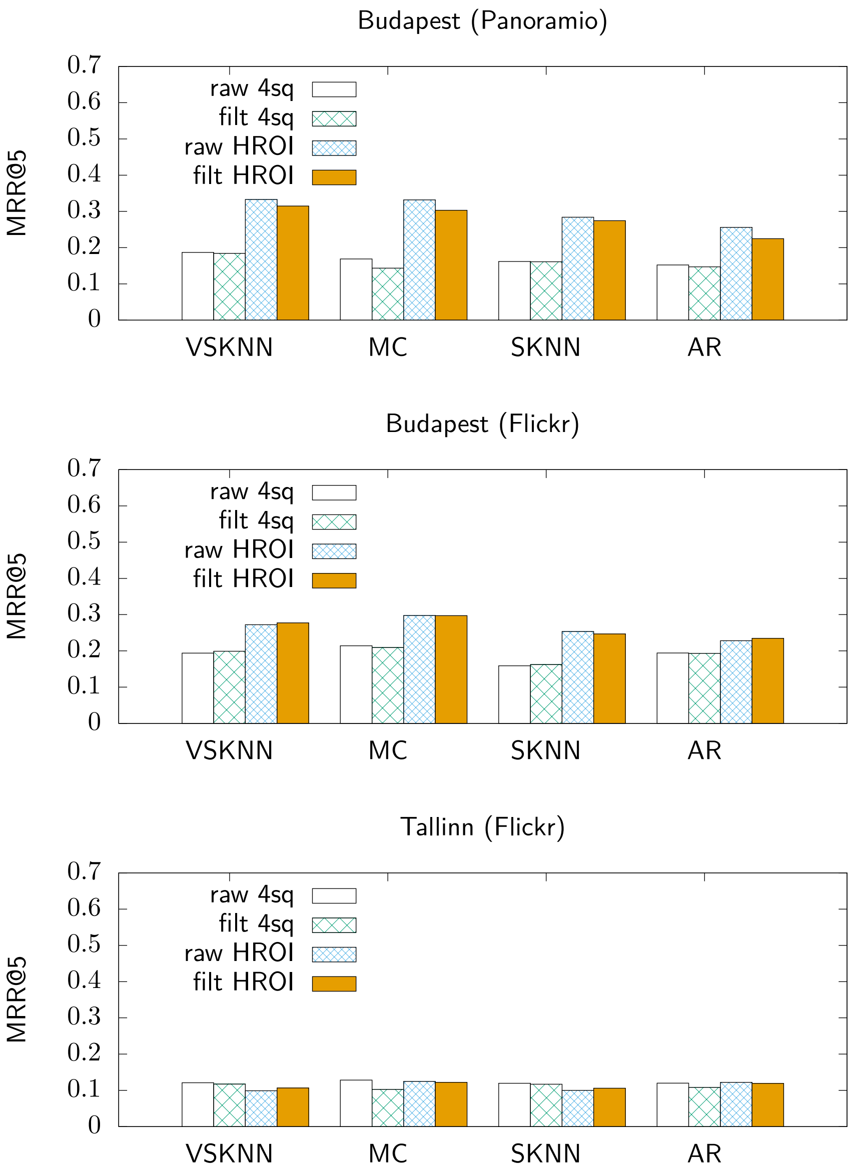

*Figure 13: Prediction accuracy with different training data preparation methods. The graphs illustrate how our proposed filtering method and annotation with hierarchical regions (HROI) impact the quality of the training data, compared to unfiltered data and annotations with a POI database. We measure the impact by training and testing with different predictive models. Compared methods: unfiltered data annotated with Foursquare POIs (*raw 4sq*); filtered data and Foursquare POIs (*filt 4sq*); unfiltered data annotated with hierarchical ROIs (*raw HROI*); filtered data and HROIs (*filt HROI*).*

Figure 13 reports the result of the experiment measuring the impact of data preparation methods on model prediction accuracy. We report the next item prediction accuracy results using the MRR@5 metric over each dataset, tested with four different models.

For the Budapest datasets, when the training data is annotated with our proposed hierarchical regions (*HROI*) the prediction accuracy is increased, compared to the POI based methods (*4sq*). However, this result is not reproduced on the Tallinn dataset. We hypothesized that this is due to lower quality region boundaries, specifically regions that are fragmented into small pieces. Since the HROI method as described in our earlier publication [53] and in this thesis does not include place name deduplication, these fragments were treated as different places. To test this hypothesis, we applied simple deduplication by merging places with exactly matching names together. The results are shown in Figure 15 and confirm that the lowered performance was caused by duplication of places.

The results of the prediction accuracy experiments imply that our proposed representation of places as hierarchical regions has a positive impact on the output of predictive models. However, the accuracy of region boundaries and naming is critical here. Both boundary and naming accuracy can be improved by place deduplication for which methods were presented in the earlier publications of our research group [77].

Filtering has no significant positive or negative impact on prediction accuracy. This could be caused by the filter failing to detect all types of noise (see Figure 6) as well as the filter being too aggressive, resulting in removal of valid training samples.

In Figure 14 we report the upper bound on the theoretical predictability for each of the training datasets. Again, the HROI annotation method gives an increase in predictability, while filtering has a small effect that fluctuates between increasing and decreasing the predictability. Due to region fragmentation in ROI extraction for Tallinn, the predictability is not increased on that dataset with the HROI method.

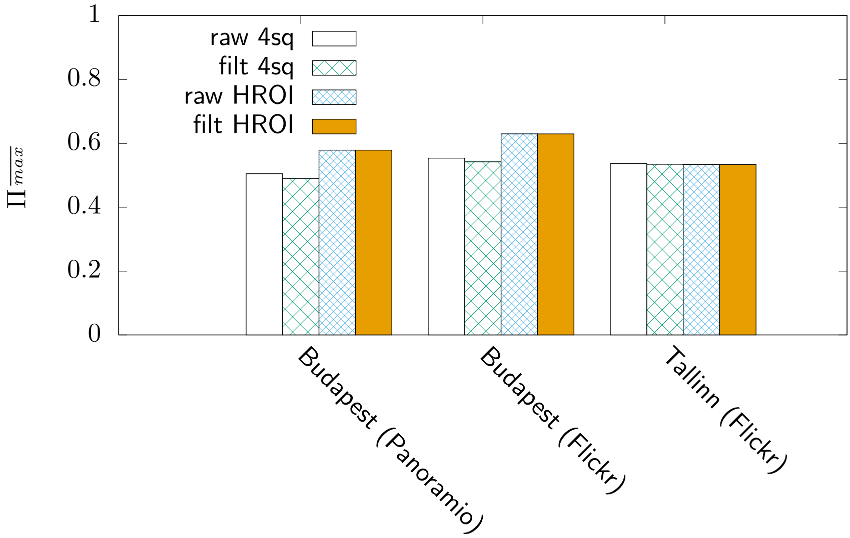

Figure 14: The impact of filtering and annotation stage on information theoretical predictability $\Pi_{\overline{max}}$ of the visit sequences derived from geo-tagged photos.

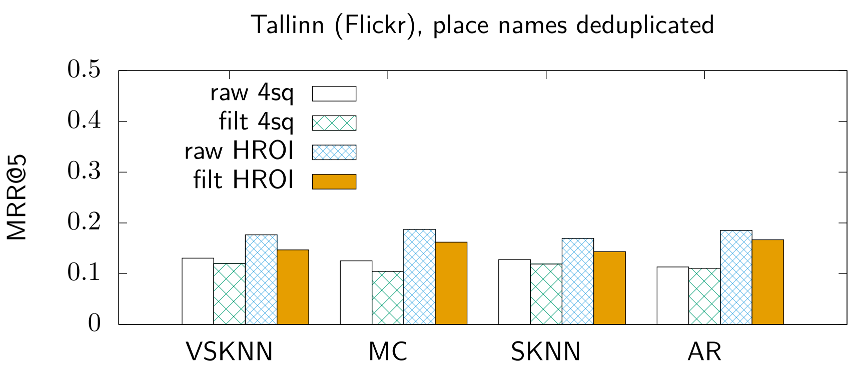

Figure 15: Comparison of data preparation methods on the Tallinn dataset using simple place deduplication. The accuracy of the predictive model is improved by deduplication when the semantic annotation is done using the hierarchical ROI method (HROI) that we developed. raw 4sq - *unfiltered data annotated with Foursquare POIs,* filt 4sq - *filtered data and Foursquare POIs,* raw HROI - *unfiltered data annotated with hierarchical ROIs,* filt HROI - *filtered data and HROIs.*

5 Conclusions

We have described a framework of place and trip recommendation systems that provides the workflow from data acquisition to recommendation (Figure 16). The workflow uses geo-tagged photos as a data source, although it is possible to use other sources in specific stages. For example, trajectories from accurate positioning devices like GPS can be used instead of traces generated from photos. This would require using appropriate stay point detection and noise filtering techniques.

We divide the workflow in two stages: knowledge engineering and creating a predictive model. The contributions of this thesis include both enhancements to the knowledge engineering stage, with the aim to improve the performance of the predictive models through higher quality training data, as well as quantitative evaluations of the methods used in the knowledge engineering stage. To the best of our knowledge, there has been no systematic study of the effects and performance of data preparation methods in the context of place recommendation.

The first step in the knowledge engineering stage is trajectory extraction. In the trajectory extraction step, we used existing methods to select tourists and their photo traces. We developed a simple threshold-based heuristic to filter erroneous data in these photo streams that uses geography-aware estimates of movement speeds. We also proposed a new clustering technique to find stay points. This technique is aware of the geographical features and therefore will not attempt to cluster together points that are separated by obstacles that prevent movement. We did small-scale quantitative evaluation of the tourist selection and our stay point detection algorithms, as well as a quantitative evaluation of the noise filtering method on a synthetic dataset.

We found the tourist selection method to be 90.2% accurate on a validation set of 234 users. Our proposed stay point detection method produced accurate clusters representing a visit to a single place in 95.7% of 1716 evaluated stay points. Because there are no baselines to compare these measurements, we cannot draw definite conclusions about whether this level accuracy is sufficient. Whether we correctly capture the behavior of our target group of users could only be measured in a user study or online evaluation.

In the experiment with synthetic data, the noise filter was very effective in detecting invalid coordinates. The filter detected 97% of the photos with invalid coordinated in the synthetic dataset. Incorrect time stamps still present a challenge. The distribution of noise is such that overall the filter detects around 60% of the noise with the settings we used in the experiments.

The second step in the knowledge engineering state is semantic annotation, where place names get assigned to stay points in trajectories. Typically, proximity to known POIs, represented as single points on the map, has been used, although the more spatially accurate representation of places as regions of interest (ROIs) has also been employed in the literature. We proposed that because the urban space forms a hierarchy of places, a layered hierarchy of regions should be used in semantic annotation. We developed new methods to extract the layered hierarchy of places represented as ROIs in a data-driven manner, as well as semantic similarity based methods to find the names for the places.

We asked over 300 residents of Tallinn to rate the ROI boundary accuracy by letting them compare region boundaries produced with single and multi-layered methods and to judge how well these match to actual geographical objects. The responses to the survey indicated that the hierarchical representation produces more accurate region boundaries.

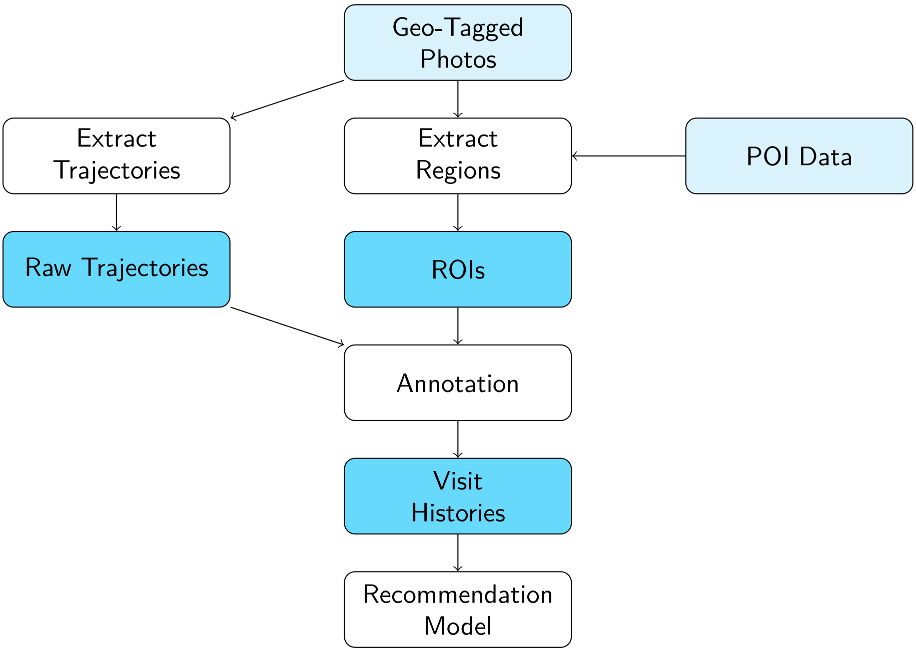

Figure 16: Architectural view of the workflow from photos to recommendation.

We evaluated the annotation accuracy of our proposed method using hierarchical regions (HROI) by comparing it to baseline POI-based methods. We separately measured accuracy on area and place level, to determine both how the addition of larger areas improves annotation and how the method performs on the finer granularity place level. The aggregate results showed that on place level (building, statue, establishment) the accuracy of HROI was close to the best POI based method. At the same time HROI increased coverage of trajectories by adding area level (park, neighborhood, island) annotations, with overall 76% of stay points receiving correct place labels. The only negative aspect of HROI was its high computational cost, caused by the choice to use a semantic distance function instead of more conventional spatial distance.

Our proposal of introducing the human expert into the ROI naming process may seem controversial in the big data era. However, we argue that pragmatic decisions are required in applications. In industry applications that integrate recommenders, it is common to use content editors and curators in various fields such as entertainment, fashion and news. Therefore the human expert should not be viewed as the band-aid for the method, but rather that the two complement each other. The automated processes can prepare and filter the data in a way to minimize the cost of the work of the human expert. The main concern with this approach is scalability when we move to a worldwide scale.

As part of the hierarchical ROI boundary extraction, we developed an extension to the existing HDBSCAN spatial clustering algorithm. The extended algorithm can automatically produce a multi-layered hierarchy that represents the most stable configuration of clusters. The stability in this context means that even if we change the requirements of how densely the data points should be located to form a cluster, the cluster boundaries change relatively little. Our proposed extension called HDBSCAN/n is a general purpose algorithm that can be applied in applications other than place recommendation or semantic annotation.

In the main evaluation experiment, we measured the overall impact of the methods we developed to recommendation. We chose the session-based scenario of recommendation that has many practical advantages in place and trip recommendation. Namely, by adopting this model we can develop applications that do not require much input from the user towards receiving recommendations. The session-based model accommodates anonymous users which is of interest in the privacy-conscious era.

We evaluated the impact of our proposed data preparation methods using prediction accuracy metrics. We found that the choice of annotation method has significant impact on the recommender performance. We achieved better prediction accuracy with training data annotated using HROI, compared to a POI-based heuristic, however this required that the cluster boundaries corresponded to the places that were typical visit destinations. In one case, cluster fragmentation required name-based merging of discovered places to get a measurable benefit from the HROI method.

We measured no significant positive effect from the noise filtering method that we proposed. The failure of the noise filter in impact evaluation can be attributed to the overall low performance of the filter - the cost of achieving approximately 60% recall resulted in the trade off of precision also dropping close to 60% on the separate synthetic dataset filter set. Assuming that the distribution of noise in the synthetic dataset is close to the real world data used in evaluation, similar numbers of both invalid and valid samples get removed and the effect becomes indistinguishable from random sampling noise. We can therefore conclude that the high detection rate of coordinate noise alone is not sufficient to have a positive impact.

We additionally measured the predictability of training and testing splits created for the impact evaluation experiments. For this, we used an information theoretic predictability estimation method that we adapted for session-based recommendation. The adapted method is model agnostic and designed specifically to evaluate datasets. The predictability measurements confirmed all the results from the model accuracy experiments: 1.) HROI improved the predictability of the datasets; 2.) HROI requires accurate region boundaries or deduplication heuristics to have impact; 3.) our proposed noise filtering method does not have positive impact.

The results of the experiments show that the hierarchical ROI representation of the urban environment we have proposed for semantic annotation is an improvement compared to naive methods in annotation accuracy, coverage of input data and predictive model accuracy. The early results we published regarding noise filtering [51] were not confirmed to have practical benefit in a different experimental setting. However, this should not be interpreted as the invalid data having no detrimental effect. Rather, the noise filtering needs to be improved.

Our research represents small steps forward in several of the stages involved in building a good model for place and trip itinerary recommendation. Substantial work remains to be done.

In noise filtering, we did not develop a good heuristic to detect cases where photo time stamps are spaced apart realistically, but the trajectories themselves are not real. For example, the user might upload travel photos slowly over a day in an arbitrary order, with the upload or editing time stamps being assigned to photos. Increasing the detection rate of such cases may be required to get a measurable benefit from noise filtering.

Semantic annotation with regions aims to represent real-world places more accurately, hence avoiding issues like designing heuristics to select the correct POI from among multiple candidates. However, the process as outlined has two variables that can both be sources for error - the region shape and region name. Additionally, the name selection is dependent on the shape. While the region shape is the intuitive representation of spatial extent of places and useful in visualization, it is not strictly necessary. Methods that do consider the spatial extent of places but do not depend on exact shape should be evaluated. One possibility is to use a grid structure similar to a quadtree where names are found for grid cells at different spatial resolution.

Perhaps the most significant shortcoming of our proposed HROI method is that the version described here does not address two important practical issues: deduplication of places and support of languages. In the one experiment we performed with deduplication, it improved the accuracy of the trained recommendation models, so there is also quantitative evidence that this step is essential in data preparation.

We also developed an algorithm to extract multi-layered clusterings from the tree produced by HDBSCAN. Since it is a general purpose spatial clustering algorithm it should be evaluated in the context of other potential applications.

The research presented in this work began before 2013 when public APIs allowed easy access to various social networks and other applications that had a large user base contributing geo-tagged data. In the following years data breaches, as well as waning interest from large data providers have caused closures and limitations of APIs. Public distrust and privacy concerns have motivated tighter legislation in terms of not only distributing, but even storing personally identifiable information. In this light we should not only ask the question, what else can we do with geo-tagged data in place recommendation, but also, how can we move further when we have limited possibilities to use this data.

List of Figures

List of Tables

References

[1] P. Adamopoulos. Beyond rating prediction accuracy: on new perspectives in recommender systems. In Q. Yang, I. King, Q. Li, P. Pu, and G. Karypis, editors, *Seventh ACM Conference on Recommender Systems, RecSys '13, Hong Kong, China, October 12-16, 2013*, pages 459–462. ACM, 2013.

[2] R. Ahas, A. Aasa, A. Roose, Ülar Mark, and S. Silm. Evaluating passive mobile positioning data for tourism surveys: An estonian case study. *Tourism Management*, 29(3):469 – 486, 2008.

[3] A. Ahmed, L. Hong, and A. J. Smola. Hierarchical geographical modeling of user locations from social media posts. In D. Schwabe, V. A. F. Almeida, H. Glaser, R. A. Baeza-Yates, and S. B. Moon, editors, *22nd International World Wide Web Conference, WWW '13, Rio de Janeiro, Brazil, May 13-17, 2013*, pages 25–36. International World Wide Web Conferences Steering Committee / ACM, 2013.

[4] L. O. Alvares, V. Bogorny, B. Kuijpers, J. A. F. de Macêdo, B. Moelans, and A. A. Vaisman. A model for enriching trajectories with semantic geographical information. In *15th ACM International Symposium on Geographic Information Systems, ACM-GIS 2007, Proceedings*, page 22, 2007.

[5] G. L. Andrienko, N. V. Andrienko, G. Fuchs, A. O. Raimond, J. Symanzik, and C. Ziemlicki. Extracting semantics of individual places from movement data by analyzing temporal patterns of visits. In S. Scheider, B. Adams, K. Janowicz, M. Vasardani, and S. Winter, editors, *ACM SIGSPATIAL International Workshop on Computational Models of Place, COMP 2013, November 5, 2013, Orlando, Florida, USA*, pages 9–15. ACM, 2013.

[6] N. V. Andrienko, G. L. Andrienko, G. Fuchs, and P. Jankowski. Scalable and privacy-respectful interactive discovery of place semantics from human mobility traces. *Information Visualization*, 15(2):117–153, 2016.

[7] M. Ankerst, M. M. Breunig, H. Kriegel, and J. Sander. OPTICS: ordering points to identify the clustering structure. In A. Delis, C. Faloutsos, and S. Ghandeharizadeh, editors, *SIGMOD 1999, Proceedings ACM SIGMOD International Conference on Management of Data, June 1-3, 1999, Philadelphia, Pennsylvania, USA.*, pages 49–60. ACM Press, 1999.

[8] Y. Arase, X. Xie, T. Hara, and S. Nishio. Mining people's trips from large scale geo-tagged photos. In A. D. Bimbo, S. Chang, and A. W. M. Smeulders, editors, *Proceedings of the 18th International Conference on Multimedia 2010, Firenze, Italy, October 25-29, 2010*, pages 133–142. ACM, 2010.

[9] G. Atluri, A. Karpatne, and V. Kumar. Spatio-temporal data mining: A survey of problems and methods. *ACM Comput. Surv.*, 51(4):83:1–83:41, 2018.

[10] D. Birant and A. Kut. ST-DBSCAN: an algorithm for clustering spatial-temporal data. *Data Knowl. Eng.*, 60(1):208–221, 2007.

[11] G. Bonnin and D. Jannach. Automated generation of music playlists: Survey and experiments. *ACM Comput. Surv.*, 47(2):26:1–26:35, 2014.

[12] I. R. Brilhante, J. A. F. de Macêdo, F. M. Nardini, R. Perego, and C. Renso. On planning sightseeing tours with tripbuilder. *Inf. Process. Manage.*, 51(2):1–15, 2015.

[13] G. Cai, C. Hio, L. Bermingham, K. Lee, and I. Lee. Mining frequent trajectory patterns and regions-of-interest from flickr photos. In *47th Hawaii International Conference on System Sciences, HICSS 2014, Waikoloa, HI, USA, January 6-9, 2014*, pages 1454–1463. IEEE Computer Society, 2014.

[14] G. Cai, C. Hio, L. Bermingham, K. Lee, and I. Lee. Sequential pattern mining of geo-tagged photos with an arbitrary regions-of-interest detection method. *Expert Syst. Appl.*, 41(7):3514–3526, 2014.

[15] R. J. G. B. Campello, D. Moulavi, and J. Sander. Density-based clustering based on hierarchical density estimates. In *Advances in Knowledge Discovery and Data Mining, 17th Pacific-Asia Conference, PAKDD 2013, Proceedings, Part II*, pages 160–172, 2013.

[16] X. Cao, G. Cong, and C. S. Jensen. Mining significant semantic locations from GPS data. *PVLDB*, 3(1):1009–1020, 2010.

[17] E. Çelikten, G. L. Falher, and M. Mathioudakis. Modeling urban behavior by mining geotagged social data. *IEEE Trans. Big Data*, 3(2):220–233, 2017.

[18] C. Chen, D. Zhang, B. Guo, X. Ma, G. Pan, and Z. Wu. Tripplanner: Personalized trip planning leveraging heterogeneous crowdsourced digital footprints. *IEEE Trans. Intelligent Transportation Systems*, 16(3):1259–1273, 2015.

[19] S. Chen, J. L. Moore, D. Turnbull, and T. Joachims. Playlist prediction via metric embedding. In Q. Yang, D. Agarwal, and J. Pei, editors, *The 18th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, KDD '12, Beijing, China, August 12-16, 2012*, pages 714–722. ACM, 2012.

[20] X. Chen, J. Pang, and R. Xue. Constructing and comparing user mobility profiles. *TWEB*, 8(4):21:1–21:25, 2014.

[21] C. Cheng, H. Yang, M. R. Lyu, and I. King. Where you like to go next: Successive point-of-interest recommendation. In F. Rossi, editor, *IJCAI 2013, Proceedings of the 23rd International Joint Conference on Artificial Intelligence, Beijing, China, August 3-9, 2013*, pages 2605–2611. IJCAI/AAAI, 2013.

[22] W. Cheng, G. Yin, Y. Dong, H. Dong, and W. Zhang. Collaborative filtering recommendation on users' interest sequences. *PloS one*, 11(5):e0155739, 2016.

[23] M. D. Choudhury, M. Feldman, S. Amer-Yahia, N. Golbandi, R. Lempel, and C. Yu. Automatic construction of travel itineraries using social breadcrumbs. In *HT'10, Proceedings of the 21st ACM Conference on Hypertext and Hypermedia*, pages 35–44, 2010.

[24] F. Christoffel, B. Paudel, C. Newell, and A. Bernstein. Blockbusters and wallflowers: Accurate, diverse, and scalable recommendations with random walks. In *Proceedings of the 9th ACM Conference on Recommender Systems*, RecSys '15, pages 163–170, New York, NY, USA, 2015. ACM.

[25] C. Comito, D. Falcone, and D. Talia. Mining human mobility patterns from social geo-tagged data. *Pervasive and Mobile Computing*, 33:91–107, 2016.

[26] D. J. Crandall, L. Backstrom, D. P. Huttenlocher, and J. M. Kleinberg. Mapping the world's photos. In *Proceedings of the 18th International Conference on World Wide Web, WWW 2009*, pages 761–770, 2009.

[27] P. Cremonesi, F. Garzotto, and R. Turrin. User-centric vs. system-centric evaluation of recommender systems. In P. Kotzé, G. Marsden, G. Lindgaard, J. Wesson, and M. Winckler, editors, *Human-Computer Interaction - INTERACT 2013 - 14th IFIP TC 13 International Conference, Cape Town, South Africa, September 2-6, 2013, Proceedings, Part III*, volume 8119 of *Lecture Notes in Computer Science*, pages 334–351. Springer, 2013.

[28] M. F. Dacrema, P. Cremonesi, and D. Jannach. Are we really making much progress? A worrying analysis of recent neural recommendation approaches. In T. Bogers, A. Said, P. Brusilovsky, and D. Tikk, editors, *Proceedings of the 13th ACM Conference on Recommender Systems, RecSys 2019, Copenhagen, Denmark, September 16-20, 2019.*, pages 101–109. ACM, 2019.

[29] S. C. Deerwester, S. T. Dumais, T. K. Landauer, G. W. Furnas, and R. A. Harshman. Indexing by latent semantic analysis. *JASIS*, 41(6):391–407, 1990.

[30] B. Donovan and D. Work. New york city taxi trip data (2010-2013), 2014.

[31] K. Drakonakis, P. Ilia, S. Ioannidis, and J. Polakis. Please forget where I was last summer: The privacy risks of public location (meta)data. In *26th Annual Network and Distributed System Security Symposium, NDSS 2019, San Diego, California, USA, February 24-27, 2019*. The Internet Society, 2019.

[32] M. Ester, H. Kriegel, J. Sander, and X. Xu. A density-based algorithm for discovering clusters in large spatial databases with noise. In E. Simoudis, J. Han, and U. M. Fayyad, editors, *Proceedings of the Second International Conference on Knowledge Discovery and Data Mining (KDD-96), Portland, Oregon, USA*, pages 226–231. AAAI Press, 1996.

[33] M. W. Fagerland, S. Lydersen, and P. Laake. The mcnemar test for binary matched-pairs data: mid-p and asymptotic are better than exact conditional. *BMC Medical Research Methodology*, 13(1):91, 2013.

[34] G. L. Falher, A. Gionis, and M. Mathioudakis. Where is the soho of rome? measures and algorithms for finding similar neighborhoods in cities. In M. Cha, C. Mascolo, and C. Sandvig, editors, *Proceedings of the Ninth International Conference on Web and Social Media, ICWSM 2015, University of Oxford, Oxford, UK, May 26-29, 2015*, pages 228–237. AAAI Press, 2015.

[35] B. Furletti, P. Cintia, C. Renso, and L. Spinsanti. Inferring human activities from GPS tracks. In Y. Zheng, S. E. Koonin, and O. E. Wolfson, editors, *Proceedings of the 2nd ACM SIGKDD International Workshop on Urban Computing, UrbComp@KDD 2013, Chicago, Illinois, USA, August 11, 2013*, pages 5:1–5:8. ACM, 2013.

[36] S. Gao, K. Janowicz, and H. Couclelis. Extracting urban functional regions from points of interest and human activities on location-based social networks. *Trans. GIS*, 21(3):446–467, 2017.

[37] D. Gavalas, C. Konstantopoulos, K. Mastakas, and G. E. Pantziou. A survey on algorithmic approaches for solving tourist trip design problems. *J. Heuristics*, 20(3):291–328, 2014.

[38] F. Giannotti, M. Nanni, D. Pedreschi, and F. Pinelli. Mining sequences with temporal annotations. In H. Haddad, editor, *Proceedings of the 2006 ACM Symposium on Applied Computing (SAC), Dijon, France, April 23-27, 2006*, pages 593–597. ACM, 2006.

[39] F. Giannotti, M. Nanni, F. Pinelli, and D. Pedreschi. Trajectory pattern mining. In *Proceedings of the 13th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 330–339, 2007.

[40] F. Girardin, F. D. Fiore, C. Ratti, and J. Blat. Leveraging explicitly disclosed location information to understand tourist dynamics: a case study. *J. Location Based Services*, 2(1):41–56, 2008.

[41] C. A. Gomez-Uribe and N. Hunt. The netflix recommender system: Algorithms, business value, and innovation. *ACM Transactions on Management Information Systems (TMIS)*, 6(4):13, 2016.

[42] A. Gunawan, H. C. Lau, and P. Vansteenwegen. Orienteering problem: A survey of recent variants, solution approaches and applications. *European Journal of Operational Research*, 255(2):315–332, 2016.

[43] G. Gupta, A. Liu, and J. Ghosh. Automated hierarchical density shaving: A robust automated clustering and visualization framework for large biological data sets. *IEEE/ACM Trans. Comput. Biology Bioinform.*, 7(2):223–237, 2010.

[44] D. L. Hansen and J. Golbeck. Mixing it up: recommending collections of items. In D. R. O. Jr., R. B. Arthur, K. Hinckley, M. R. Morris, S. E. Hudson, and S. Greenberg, editors, *Proceedings of the 27th International Conference on Human Factors in Computing Systems, CHI 2009, Boston, MA, USA, April 4-9, 2009*, pages 1217–1226. ACM, 2009.

[45] B. J. Hecht and M. Stephens. A tale of cities: Urban biases in volunteered geographic information. In E. Adar, P. Resnick, M. D. Choudhury, B. Hogan, and A. H. Oh, editors, *Proceedings of the Eighth International Conference on Weblogs and Social Media, ICWSM 2014, Ann Arbor, Michigan, USA, June 1-4, 2014.* The AAAI Press, 2014.

[46] D. Herzog and W. Wörndl. User-centered evaluation of strategies for recommending sequences of points of interest to groups. In T. Bogers, A. Said, P. Brusilovsky, and D. Tikk, editors, *Proceedings of the 13th ACM Conference on Recommender Systems, RecSys 2019, Copenhagen, Denmark, September 16-20, 2019.*, pages 96–100. ACM, 2019.

[47] B. Hidasi and A. Karatzoglou. Recurrent neural networks with top-k gains for session-based recommendations. In A. Cuzzocrea, J. Allan, N. W. Paton,

D. Srivastava, R. Agrawal, A. Z. Broder, M. J. Zaki, K. S. Candan, A. Labrinidis, A. Schuster, and H. Wang, editors, *Proceedings of the 27th ACM International Conference on Information and Knowledge Management, CIKM 2018, Torino, Italy, October 22-26, 2018*, pages 843–852. ACM, 2018.

[48] N. Hurley and M. Zhang. Novelty and diversity in top-n recommendation - analysis and evaluation. *ACM Trans. Internet Techn.*, 10(4):14, 2011.

[49] D. Jannach, I. Kamehkhosh, and L. Lerche. Leveraging multi-dimensional user models for personalized next-track music recommendation. In A. Seffah, B. Penzenstadler, C. Alves, and X. Peng, editors, *Proceedings of the Symposium on Applied Computing, SAC 2017, Marrakech, Morocco, April 3-7, 2017*, pages 1635–1642. ACM, 2017.

[50] D. Jannach, L. Lerche, I. Kamehkhosh, and M. Jugovac. What recommenders recommend: an analysis of recommendation biases and possible countermeasures. *User Modeling and User-Adapted Interaction*, 25(5):427–491, 2015.

[51] P. Järv. Extracting human mobility data from geo-tagged photos. In *Proceedings of the PredictGIS'17:1st ACM SIGSPATIAL Workshop on Prediction of Human Mobility*, pages 4:1–4:7, 2017.

[52] P. Järv. Predictability limits in session-based next item recommendation. In T. Bogers, A. Said, P. Brusilovsky, and D. Tikk, editors, *Proceedings of the 13th ACM Conference on Recommender Systems, RecSys 2019, Copenhagen, Denmark, September 16-20, 2019.*, pages 146–150. ACM, 2019.

[53] P. Järv, T. Tammet, and M. Tall. Hierarchical regions of interest. In *19th IEEE International Conference on Mobile Data Management, MDM 2018, Aalborg, Denmark, June 25-28, 2018*, pages 86–95. IEEE Computer Society, 2018.

[54] S. Jiang, J. F. Jr., and M. C. González. Activity-based human mobility patterns inferred from mobile phone data: A case study of singapore. *IEEE Trans. Big Data*, 3(2):208–219, 2017.

[55] S. Jiang, X. Qian, T. Mei, and Y. Fu. Personalized travel sequence recommendation on multi-source big social media. *IEEE Trans. Big Data*, 2(1):43–56, 2016.

[56] S. Jiang, Y. Yang, S. Gupta, D. Veneziano, S. Athavale, and M. C. González. The timegeo modeling framework for urban mobility without travel surveys. *Proceedings of the National Academy of Sciences*, 113(37):E5370–E5378, 2016.

[57] S. Kisilevich, F. Mansmann, and D. A. Keim. P-DBSCAN: a density based clustering algorithm for exploration and analysis of attractive areas using collections of geo-tagged photos. In *Proceedings of the 1st International Conference and Exhibition on Computing for Geospatial Research & Application, COM.Geo 2010*, 2010.

[58] J. A. Konstan and J. Riedl. Recommender systems: from algorithms to user experience. *User Model. User-Adapt. Interact.*, 22(1-2):101–123, 2012.

[59] V. Kulkarni, A. Moro, and B. Garbinato. Mobidict: a mobility prediction system leveraging realtime location data streams. In F. B. Kashani, C. Zhang, and A. M. Hendawi, editors, *Proceedings of the 7th ACM SIGSPATIAL International Workshop on GeoStreaming, IWGS@SIGSPATIAL 2016, Burlingame, California, USA, October 31 - November 3, 2016*, pages 8:1–8:10. ACM, 2016.

[60] T. Kurashima, T. Iwata, T. Hoshide, N. Takaya, and K. Fujimura. Geo topic model: joint modeling of user's activity area and interests for location recommendation. In S. Leonardi, A. Panconesi, P. Ferragina, and A. Gionis, editors, *Sixth ACM International Conference on Web Search and Data Mining, WSDM 2013, Rome, Italy, February 4-8, 2013*, pages 375–384. ACM, 2013.

[61] T. Kurashima, T. Iwata, G. Irie, and K. Fujimura. Travel route recommendation using geotags in photo sharing sites. In J. Huang, N. Koudas, G. J. F. Jones, X. Wu, K. Collins-Thompson, and A. An, editors, *Proceedings of the 19th ACM Conference on Information and Knowledge Management, CIKM 2010, Toronto, Ontario, Canada, October 26-30, 2010*, pages 579–588. ACM, 2010.

[62] D. Laptev, A. Tikhonov, P. Serdyukov, and G. Gusev. Parameter-free discovery and recommendation of areas-of-interest. In *Proceedings of the 22nd ACM SIGSPATIAL International Conference on Advances in Geographic Information Systems*, pages 113–122, 2014.

[63] H. Laurmaa. Möödunud aasta oli turismis rekordiline. Technical report, Eesti Statistikaamet, 2018.

[64] C. Li, Y. Zhao, X. Sun, X. Su, S. Zheng, R. Dong, and L. Shi. Photography-based analysis of tourists' temporal–spatial behaviour in the old town of lijiang. *International Journal of Sustainable Development & World Ecology*, 18(6):523–529, 2011.

[65] M. Li, R. Westerholt, H. Fan, and A. Zipf. Assessing spatiotemporal predictability of lbsn: a case study of three foursquare datasets. *GeoInformatica*, pages 1–21, 2016.

[66] Q. Li, Y. Zheng, X. Xie, Y. Chen, W. Liu, and W. Ma. Mining user similarity based on location history. In W. G. Aref, M. F. Mokbel, and M. Schneider, editors, *16th ACM SIGSPATIAL International Symposium on Advances in Geographic Information Systems, ACM-GIS 2008, November 5-7, 2008, Irvine, California, USA, Proceedings*, page 34. ACM, 2008.

[67] L. Liao, D. Fox, and H. A. Kautz. Extracting places and activities from GPS traces using hierarchical conditional random fields. *I. J. Robotics Res.*, 26(1):119–134, 2007.

[68] K. H. Lim, J. Chan, S. Karunasekera, and C. Leckie. Tour recommendation and trip planning using location-based social media: a survey. *Knowl. Inf. Syst.*, 60(3):1247–1275, 2019.

[69] K. H. Lim, J. Chan, C. Leckie, and S. Karunasekera. Personalized tour recommendation based on user interests and points of interest visit durations. In *Proceedings of the Twenty-Fourth International Joint Conference on Artificial Intelligence, IJCAI 2015*, pages 1778–1784, 2015.

[70] A. Lima, R. Stanojevic, D. Papagiannaki, P. Rodriguez, and M. C. González. Understanding individual routing behaviour. *Journal of The Royal Society Interface*, 13(116), 2016.

[71] G. Linden, B. Smith, and J. York. Amazon. com recommendations: Item-to-item collaborative filtering. *IEEE Internet computing*, (1):76–80, 2003.

[72] C. Liu, K. Zhang, H. Xiong, G. Jiang, and Q. Yang. Temporal skeletonization on sequential data: Patterns, categorization, and visualization. *IEEE Trans. Knowl. Data Eng.*, 28(1):211–223, 2016.

[73] J. Liu, Z. Huang, L. Chen, H. T. Shen, and Z. Yan. Discovering areas of interest with geo-tagged images and check-ins. In *Proceedings of the 20th ACM Multimedia Conference, MM '12*, pages 589–598, 2012.

[74] S. Liu, M. Araujo, E. Brunskill, R. J. F. Rossetti, J. Barros, and R. Krishnan. Understanding sequential decisions via inverse reinforcement learning. In *2013 IEEE 14th International Conference on Mobile Data Management, Milan, Italy, June 3-6, 2013 - Volume 1*, pages 177–186. IEEE Computer Society, 2013.

[75] Y. Liu, C. Liu, B. Liu, M. Qu, and H. Xiong. Unified point-of-interest recommendation with temporal interval assessment. In B. Krishnapuram, M. Shah, A. J. Smola, C. C. Aggarwal, D. Shen, and R. Rastogi, editors, *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, San Francisco, CA, USA, August 13-17, 2016*, pages 1015–1024. ACM, 2016.

[76] X. Lu, C. Wang, J. Yang, Y. Pang, and L. Zhang. Photo2trip: generating travel routes from geo-tagged photos for trip planning. In *Proceedings of the 18th International Conference on Multimedia 2010*, pages 143–152, 2010.

[77] A. Luberg, M. Granitzer, H. Wu, P. Järv, and T. Tammet. Information retrieval and deduplication for tourism recommender sightsplanner. In D. D. Burdescu, R. Akerkar, and C. Badica, editors, *2nd International Conference on Web Intelligence, Mining and Semantics, WIMS '12, Craiova, Romania, June 6-8, 2012*, pages 50:1–50:11. ACM, 2012.

[78] A. Luberg, P. Järv, K. Schoefegger, and T. Tammet. Context-aware and multilingual information extraction for a tourist recommender system. In S. N. Lindstaedt and M. Granitzer, editors, *I-KNOW 2011, 11th International Conference on Knowledge Management and Knowledge Technologies, Graz, Austria, September 7-9, 2011*, page 13. ACM, 2011.

[79] A. Luberg, P. Järv, and T. Tammet. Information extraction for a tourist recommender system. In M. Fuchs, F. Ricci, and L. Cantoni, editors, *Information and Communication Technologies in Tourism 2012, ENTER 2012, Proceedings of the International Conference in Helsingborg, Sweden, January 25-27, 2012.*, pages 332–343. Springer, 2012.

[80] A. Luberg, P. Järv, and T. Tammet. Information extraction for a tourist recommender system. In M. Fuchs, F. Ricci, and L. Cantoni, editors, *Information and Communication Technologies in Tourism 2012*, pages 332–343. Springer Vienna, 2012.

[81] A. Luberg, T. Tammet, and P. Järv. Extended triple store structure used in recommender system. In F. Morvan, A. M. Tjoa, and R. Wagner, editors, *2011 Database and Expert Systems Applications, DEXA, International Workshops, Toulouse, France, August 29 - Sept. 2, 2011*, pages 539–543. IEEE Computer Society, 2011.

[82] A. Luberg, T. Tammet, and P. Järv. Smart city: A rule-based tourist recommendation system. In R. Law, M. Fuchs, and F. Ricci, editors, *Information and Communication Technologies in Tourism 2011 - Proceedings of the International Conference in Innsbruck, Austria, January 26-28, 2011*, pages 51–62. Springer Vienna, 2011.

[83] M. Ludewig and D. Jannach. Evaluation of session-based recommendation algorithms. *User Model. User-Adapt. Interact.*, 28(4-5):331–390, 2018.

[84] M. Ludewig, N. Mauro, S. Latifi, and D. Jannach. Performance comparison of neural and non-neural approaches to session-based recommendation. In T. Bogers, A. Said, P. Brusilovsky, and D. Tikk, editors, *Proceedings of the 13th ACM Conference on Recommender Systems, RecSys 2019, Copenhagen, Denmark, September 16-20, 2019.*, pages 462–466. ACM, 2019.

[85] D. Luxen and C. Vetter. Real-time routing with openstreetmap data. In *Proceedings of the 19th ACM SIGSPATIAL International Conference on Advances in Geographic Information Systems*, pages 513–516. ACM, 2011.

[86] T. N. Maeda, K. Tsubouchi, and F. Toriumi. Next place prediction in unfamiliar places considering contextual factors. In *Proceedings of the 25th ACM SIGSPATIAL International Conference on Advances in Geographic Information Systems*, pages 76:1–76:4, 2017.

[87] T. N. Maeda, M. Yoshida, F. Toriumi, and H. Ohashi. Extraction of tourist destinations and comparative analysis of preferences between foreign tourists and domestic tourists on the basis of geotagged social media data. *ISPRS Int. J. Geo-Information*, 7(3):99, 2018.

[88] M. K. E. Mahrsi, E. Côme, L. Oukhellou, and M. Verleysen. Clustering smart card data for urban mobility analysis. *IEEE Trans. Intelligent Transportation Systems*, 18(3):712–728, 2017.

[89] A. Majid, L. Chen, G. Chen, H. T. Mirza, I. Hussain, and J. Woodward. A context-aware personalized travel recommendation system based on geotagged social media data mining. *International Journal of Geographical Information Science*, 27(4):662–684, 2013.

[90] M. Malik, H. Lamba, C. Nakos, and J. Pfeffer. Population bias in geotagged tweets, 2015.

[91] J. Masthoff. Group recommender systems: Combining individual models. In F. Ricci, L. Rokach, B. Shapira, and P. B. Kantor, editors, *Recommender Systems Handbook*, pages 677–702. Springer, 2011.

[92] J. Masthoff and A. Gatt. In pursuit of satisfaction and the prevention of embarrassment: affective state in group recommender systems. *User Model. User-Adapt. Interact.*, 16(3-4):281–319, 2006.

[93] J. D. Mazimpaka and S. Timpf. Trajectory data mining: A review of methods and applications. *Journal of Spatial Information Science*, 2016(13):61–99, 2016.

[94] S. M. McNee, J. Riedl, and J. A. Konstan. Being accurate is not enough: how accuracy metrics have hurt recommender systems. In G. M. Olson and R. Jeffries, editors, *Extended Abstracts Proceedings of the 2006 Conference on Human Factors in Computing Systems, CHI 2006, Montréal, Québec, Canada, April 22-27, 2006*, pages 1097–1101. ACM, 2006.

[95] A. Monreale, F. Pinelli, R. Trasarti, and F. Giannotti. Wherenext: a location predictor on trajectory pattern mining. In J. F. E. IV, F. Fogelman-Soulié, P. A. Flach, and M. J. Zaki, editors, *Proceedings of the 15th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, Paris, France, June 28 - July 1, 2009*, pages 637–646. ACM, 2009.

[96] C. I. Muntean, F. M. Nardini, F. Silvestri, and R. Baraglia. On learning prediction models for tourists paths. *ACM TIST*, 7(1):8:1–8:34, 2015.

[97] A. Noulas, S. Scellato, N. Lathia, and C. Mascolo. Mining user mobility features for next place prediction in location-based services. In M. J. Zaki, A. Siebes, J. X. Yu, B. Goethals, G. I. Webb, and X. Wu, editors, *12th IEEE International Conference on Data Mining, ICDM 2012, Brussels, Belgium, December 10-13, 2012*, pages 1038–1043. IEEE Computer Society, 2012.

[98] A. T. Palma, V. Bogorny, B. Kuijpers, and L. O. Alvares. A clustering-based approach for discovering interesting places in trajectories. In R. L. Wainwright and H. Haddad, editors, *Proceedings of the 2008 ACM Symposium on Applied Computing (SAC), Fortaleza, Ceara, Brazil, March 16-20, 2008*, pages 863–868. ACM, 2008.

[99] C. Parent et al. Semantic trajectories modeling and analysis. *ACM Comput. Surv.*, 45(4):42:1–42:32, 2013.

[100] F. Pianese, X. An, F. Kawsar, and H. Ishizuka. Discovering and predicting user routines by differential analysis of social network traces. In *IEEE 14th International Symposium on "A World of Wireless, Mobile and Multimedia Networks", WoWMoM 2013, Madrid, Spain, June 4-7, 2013*, pages 1–9. IEEE Computer Society, 2013.

[101] M. Quadrana, P. Cremonesi, and D. Jannach. Sequence-aware recommender systems. *ACM Comput. Surv.*, 51(4):66:1–66:36, 2018.

[102] D. Quercia, R. Schifanella, and L. M. Aiello. The shortest path to happiness: recommending beautiful, quiet, and happy routes in the city. In L. Ferres, G. Rossi, V. A. F. Almeida, and E. Herder, editors, *25th ACM Conference on Hypertext and Social Media, HT '14, Santiago, Chile, September 1-4, 2014*, pages 116–125. ACM, 2014.

[103] S. Rendle, C. Freudenthaler, and L. Schmidt-Thieme. Factorizing personalized markov chains for next-basket recommendation. In M. Rappa, P. Jones, J. Freire, and S. Chakrabarti, editors, *Proceedings of the 19th International Conference on World Wide Web, WWW 2010, Raleigh, North Carolina, USA, April 26-30, 2010*, pages 811–820. ACM, 2010.

[104] S. Rendle, L. Zhang, and Y. Koren. On the difficulty of evaluating baselines: A study on recommender systems. *CoRR*, abs/1905.01395, 2019.

[105] D. Reynard and M. Shirgaokar. Harnessing the power of machine learning: Can twitter data be useful in guiding resource allocation decisions during a natural disaster? *Transportation Research Part D: Transport and Environment*, 2019.

[106] J. Sander, X. Qin, Z. Lu, N. Niu, and A. Kovarsky. Automatic extraction of clusters from hierarchical clustering representations. In *Advances in Knowledge Discovery and Data Mining, 7th Pacific-Asia Conference, PAKDD 2003, Proceedings*, pages 75–87, 2003.

[107] J. Sang, T. Mei, J. Sun, C. Xu, and S. Li. Probabilistic sequential pois recommendation via check-in data. In I. F. Cruz, C. A. Knoblock, P. Kröger, E. Tanin, and P. Widmayer, editors, *SIGSPATIAL 2012 International Conference on Advances in Geographic Information Systems (formerly known as GIS), SIGSPATIAL'12, Redondo Beach, CA, USA, November 7-9, 2012*, pages 402–405. ACM, 2012.

[108] M. Shirai, M. Hirota, S. Yokoyama, N. Fukuta, and H. Ishikawa. Discovering multiple hotspots using geo-tagged photographs. In *SIGSPATIAL 2012 International Conference on Advances in Geographic Information Systems*, pages 490–493, 2012.

[109] G. Skoumas, K. A. Schmid, G. Jossé, M. Schubert, M. A. Nascimento, A. Züfle, M. Renz, and D. Pfoser. Knowledge-enriched route computation. In C. Claramunt, M. Schneider, R. C. Wong, L. Xiong, W. Loh, C. Shahabi, and K. Li, editors, *Advances in Spatial and Temporal Databases - 14th International Symposium, SSTD 2015, Hong Kong, China, August 26-28, 2015. Proceedings*, volume 9239 of *Lecture Notes in Computer Science*, pages 157–176. Springer, 2015.

[110] G. Smith, R. Wieser, J. Goulding, and D. Barrack. A refined limit on the predictability of human mobility. In *IEEE International Conference on Pervasive Computing and Communications, PerCom 2014, Budapest, Hungary, March 24-28, 2014*, pages 88–94. IEEE Computer Society, 2014.

[111] B. Smyth and P. McClave. Similarity vs. diversity. In D. Aha and I. Watson, editors, *Case-Based Reasoning Research and Development*, volume 2080 of *Lecture Notes in Computer Science*, pages 347–361. Springer Berlin Heidelberg, 2001.

[112] C. Song, Z. Qu, N. Blumm, and A.-L. Barabási. Limits of predictability in human mobility. *Science*, 327(5968):1018–1021, 2010.

[113] K. Swearingen and R. Sinha. Beyond algorithms: An hci perspective on recommender systems. In *ACM SIGIR 2001 Workshop on Recommender Systems*, volume 13, pages 1–11. Citeseer, 2001.

[114] T. Tammet, A. Luberg, and P. Järv. Sightsmap: Crowd-sourced popularity of the world places. In L. Cantoni and Z. P. Xiang, editors, *Information and Communication Technologies in Tourism 2013*, pages 314–325. Springer Berlin Heidelberg, 2013.

[115] K. Tomingas, P. Järv, and T. Tammet. Computing data lineage and business semantics for data warehouse. In A. L. N. Fred, J. L. G. Dietz, D. Aveiro, K. Liu, J. Bernardino, and J. Filipe, editors, *Knowledge Discovery, Knowledge Engineering and Knowledge Management - 8th International Joint Conference, IC3K 2016, Porto, Portugal, November 9-11, 2016, Revised Selected Papers*, volume 914 of *Communications in Computer and Information Science*, pages 101–124. Springer, 2016.

[116] K. Tomingas, P. Järv, and T. Tammet. Discovering data lineage from data warehouse procedures. In A. L. N. Fred, J. L. G. Dietz, D. Aveiro, K. Liu, J. Bernardino, and J. Filipe, editors, *Proceedings of the 8th International Joint Conference on Knowledge Discovery, Knowledge Engineering and Knowledge Management (IC3K 2016) - Volume 1: KDIR, Porto - Portugal, November 9 - 11, 2016.*, pages 101–110. SciTePress, 2016.

[117] K. Tomingas, T. Tammet, M. Kliimask, and P. Järv. Automating component dependency analysis for enterprise business intelligence. In M. D. Myers and D. W. Straub, editors, *Proceedings of the International Conference on Information Systems - Building a Better World through Information Systems, ICIS 2014, Auckland, New Zealand, December 14-17, 2014*. Association for Information Systems, 2014.

[118] R. Turrin, A. Condorelli, P. Cremonesi, R. Pagano, and M. Quadrana. Large scale music recommendation. In *Workshop on Large-Scale Recommender Systems (LSRS 2015) at ACM RecSys*, 2015.

[119] M. R. Uddin, C. V. Ravishankar, and V. J. Tsotras. Finding regions of interest from trajectory data. In *12th IEEE International Conference on Mobile Data Management, MDM 2011, Volume 1*, pages 39–48, 2011.

[120] J. Urner, D. Bucher, J. Yang, and D. Jonietz. Assessing the influence of spatio-temporal context for next place prediction using different machine learning approaches. *ISPRS Int. J. Geo-Information*, 7(5):166, 2018.

[121] M. Van Setten, S. Pokraev, and J. Koolwaaij. Context-aware recommendations in the mobile tourist application compass. In *International Conference on Adaptive Hypermedia and Adaptive Web-Based Systems*, pages 235–244. Springer, 2004.

[122] P. Vansteenwegen and D. V. Oudheusden. The mobile tourist guide: An OR opportunity. *OR Insight*, 20(3):21–27, 2007.

[123] S. Vargas and P. Castells. Rank and relevance in novelty and diversity metrics for recommender systems. In B. Mobasher, R. D. Burke, D. Jannach, and G. Adomavicius, editors, *Proceedings of the 2011 ACM Conference on Recommender Systems, RecSys 2011, Chicago, IL, USA, October 23-27, 2011*, pages 109–116. ACM, 2011.

[124] H. Q. Vu, G. Li, R. Law, and B. H. Ye. Exploring the travel behaviors of inbound tourists to hong kong using geotagged photos. *Tourism Management*, 46:222 – 232, 2015.

[125] G. Walsh and J. Golbeck. Curator: a game with a purpose for collection recommendation. In E. D. Mynatt, D. Schoner, G. Fitzpatrick, S. E. Hudson,

W. K. Edwards, and T. Rodden, editors, *Proceedings of the 28th International Conference on Human Factors in Computing Systems, CHI 2010, Atlanta, Georgia, USA, April 10-15, 2010*, pages 2079–2082. ACM, 2010.

[126] P. Wang, J. Guo, Y. Lan, J. Xu, S. Wan, and X. Cheng. Learning hierarchical representation model for nextbasket recommendation. In R. A. Baeza-Yates, M. Lalmas, A. Moffat, and B. A. Ribeiro-Neto, editors, *Proceedings of the 38th International ACM SIGIR Conference on Research and Development in Information Retrieval, Santiago, Chile, August 9-13, 2015*, pages 403–412. ACM, 2015.

[127] Z. Wang, S. Y. He, and Y. Leung. Applying mobile phone data to travel behaviour research: A literature review. *Travel Behaviour and Society*, 11:141 – 155, 2018.

[128] P. Xu, L. Yin, Z. Yue, and T. Zhou. On predictability of time series. *Physica A: Statistical Mechanics and its Applications*, 523:345–351, 2019.

[129] Z. Xu, L. Chen, and G. Chen. Topic based context-aware travel recommendation method exploiting geotagged photos. *Neurocomputing*, 155:99–107, 2015.

[130] Z. Yan, C. Parent, S. Spaccapietra, and D. Chakraborty. A hybrid model and computing platform for spatio-semantic trajectories. In *The Semantic Web: Research and Applications, 7th Extended Semantic Web Conference, ESWC 2010, Proceedings, Part I*, pages 60–75, 2010.

[131] Z. Yin, L. Cao, J. Han, J. Luo, and T. S. Huang. Diversified trajectory pattern ranking in geo-tagged social media. In *Proceedings of the Eleventh SIAM International Conference on Data Mining, SDM 2011*, pages 980–991, 2011.

[132] H. Ying, F. Zhuang, F. Zhang, Y. Liu, G. Xu, X. Xie, H. Xiong, and J. Wu. Sequential recommender system based on hierarchical attention networks. In J. Lang, editor, *Proceedings of the Twenty-Seventh International Joint Conference on Artificial Intelligence, IJCAI 2018, July 13-19, 2018, Stockholm, Sweden.*, pages 3926–3932. ijcai.org, 2018.

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

[134] C. Yu, L. V. S. Lakshmanan, and S. Amer-Yahia. It takes variety to make a world: diversification in recommender systems. In M. L. Kersten, B. Novikov, J. Teubner, V. Polutin, and S. Manegold, editors, *EDBT 2009, 12th International Conference on Extending Database Technology, Saint Petersburg, Russia, March 24-26, 2009, Proceedings*, volume 360 of *ACM International Conference Proceeding Series*, pages 368–378. ACM, 2009.

[135] J. Yuan, Y. Zheng, and X. Xie. Discovering regions of different functions in a city using human mobility and pois. In Q. Yang, D. Agarwal, and J. Pei, editors, *The 18th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, KDD '12, Beijing, China, August 12-16, 2012*, pages 186–194. ACM, 2012.

[136] N. J. Yuan, Y. Zheng, L. Zhang, and X. Xie. T-finder: A recommender system for finding passengers and vacant taxis. *IEEE Trans. Knowl. Data Eng.*, 25(10):2390–2403, 2013.

[137] C. Zhang, J. Han, L. Shou, J. Lu, and T. F. L. Porta. Splitter: Mining fine-grained sequential patterns in semantic trajectories. *PVLDB*, 7(9):769–780, 2014.

[138] C. Zhang, K. Zhang, Q. Yuan, H. Peng, Y. Zheng, T. Hanratty, S. Wang, and J. Han. Regions, periods, activities: Uncovering urban dynamics via cross-modal representation learning. In R. Barrett, R. Cummings, E. Agichtein, and E. Gabrilovich, editors, *Proceedings of the 26th International Conference on World Wide Web, WWW 2017, Perth, Australia, April 3-7, 2017*, pages 361–370. ACM, 2017.

[139] S. Zhang, L. Yao, A. Sun, and Y. Tay. Deep learning based recommender system: A survey and new perspectives. *ACM Comput. Surv.*, 52(1):5:1–5:38, 2019.

[140] Z. Zhang, L. Zhou, X. Zhao, G. Wang, Y. Su, M. J. Metzger, H. Zheng, and B. Y. Zhao. On the validity of geosocial mobility traces. In D. Levine, S. Katti, and D. Oran, editors, *Twelfth ACM Workshop on Hot Topics in Networks, HotNets-XII, College Park, MD, USA, November 21-22, 2013*, pages 11:1–11:7. ACM, 2013.

[141] K. Zhao, G. Cong, and A. Sun. Annotating points of interest with geo-tagged tweets. In S. Mukhopadhyay, C. Zhai, E. Bertino, F. Crestani, J. Mostafa, J. Tang, L. Si, X. Zhou, Y. Chang, Y. Li, and P. Sondhi, editors, *Proceedings of the 25th ACM International Conference on Information and Knowledge Management, CIKM 2016, Indianapolis, IN, USA, October 24-28, 2016*, pages 417–426. ACM, 2016.

[142] S. Zhao, I. King, and M. R. Lyu. A survey of point-of-interest recommendation in location-based social networks. *CoRR*, abs/1607.00647, 2016.

[143] E. Zheleva, J. Guiver, E. M. Rodrigues, and N. Milic-Frayling. Statistical models of music-listening sessions in social media. In M. Rappa, P. Jones, J. Freire, and S. Chakrabarti, editors, *Proceedings of the 19th International Conference on World Wide Web, WWW 2010, Raleigh, North Carolina, USA, April 26-30, 2010*, pages 1019–1028. ACM, 2010.

[144] Y. Zheng. Trajectory data mining: An overview. *ACM TIST*, 6(3):29:1–29:41, 2015.

[145] Y. Zheng, L. Zhang, Z. Ma, X. Xie, and W. Ma. Recommending friends and locations based on individual location history. *TWEB*, 5(1):5:1–5:44, 2011.

[146] T. Zhou, Z. Kuscsik, J.-G. Liu, M. Medo, J. R. Wakeling, Y.-C. Zhang, and G. Parisi. Solving the apparent diversity-accuracy dilemma of recommender systems. *Proceedings of the National Academy of Sciences of the United States of America*, 107(10):4511–4515, 2010.

[147] C. Ziegler, S. M. McNee, J. A. Konstan, and G. Lausen. Improving recommendation lists through topic diversification. In A. Ellis and T. Hagino, editors, *Proceedings of the 14th international conference on World Wide Web, WWW 2005, Chiba, Japan, May 10-14, 2005*, pages 22–32. ACM, 2005.

Acknowledgments

I would like to thank my supervisor, Prof. Tanel Tammet, for encouraging me to take up the PhD studies, for motivating me to continue with it throughout and for numerous enlightening discussions both related and unrelated to my research. I am also grateful to all of my colleagues at the Institute of Software Science who have supported my studies, but particularly Assoc. Prof. Juhan Ernits. Finally, writing this thesis would not have been possible without the seemingly unlimited patience of my family.

Part of the experiments were performed using hardware contributed by the HITSA Foundation. The conference visits were funded by the Dora Plus program that is supported by the EU Regional Development Fund.

Abstract
Place Recommendation with Geo-tagged Photos

Place and trip itinerary recommenders assist users in planning visits to unfamiliar destinations. Unlike in online retail and media, where users interact with applications and data about consumption is accumulated directly, data about place visits is not readily available. To build recommender models, data acquisition methods need to be developed. We focus on the implicit place visit histories in geo-tagged photos.

The user photos are sparse in both space and time and the way the data is created introduces specific types of noise. We describe trajectory extraction with awareness regarding these issues. To determine place names, we introduce a hierarchical representation of areas and places that captures their shapes and sizes, called hierarchical regions of interest (HROI). We mine region data by combining geo-tagged photos and Foursquare venues.

We approach the recommendation of places and trips as a sequential recommendation problem, where the order of places visited is considered important. Also, we adopt a session-based model that places low demands on the amount of data required from the user to make recommendations. Within this context, we evaluated the impact of the proposed methods, compared to naive approaches. The experiments confirmed that recommenders trained with data prepared by the HROI method had better prediction accuracy. We also confirmed that the HROI method has approximately similar annotation accuracy compared to the best POI based method, but annotates more places by including larger areas. These results mean that semantic annotation using ROIs is an improvement compared to the commonly used POI based methods. We were unable to confirm any positive effect from filtering, suggesting that further development of the filtering technique is needed.

Kokkuvõte
Geomärgistega fotode kaevandamine reisisoovitusteks

Reisisoovitussüsteemid aitavad kasutajatel planeerida reise sihtkohtadesse, mida kasutajad ise piisavalt ei tunne. Erinevalt veebipoodidest ja Interneti vahendusel meedia tarbimisest ei teki turismirakenduste kasutamise käigus tingimata andmeid kasutajate tegelike külastuste ja eelistuste kohta. Seetõttu vajatakse soovitusmudelite koostamiseks eraldi andmekaevemeetodeid. Käesolevas doktoritöös keskendutakse geomärgistega fotode kaevandamisele kasutajate käitumise modelleerimiseks.

Fotod paiknevad ajas ja ruumis hõredalt ning viisid, kuidas neid märgendatakse, põhjustavad asukoha- ja ajaandmetes vigu. Kirjeldame kasutajate liikumiste ja külastuste kaevandamise viise, mis arvestavad nimetatud probleemidega ning suudavad filtreerida vigadest põhjustatud müra. Kasutajate tegevuste kirjeldamiseks on vaja neid seostada kohtade nimedega, mille leidmiseks esitame hierarhilise esitusviisi linnaruumi kirjeldamiseks, kus kasutatakse kohtade esitusena erineva suurusega regioone. Regioonide leidmiseks analüüsime geomärgendatud fotode tihedust ning nende seost Foursquare andmebaasis kirjeldatud kohtadega.

Kohtade ja reisikavade soovitamine on ülesanne, kus kohtade ja tegevuste järjekord on oluline. Me eeldame ka sessioonipõhist soovitusmudelit, mis ei nõua kasutajate poolt eelistuste sisestamist ega kasutajate isikustamist. Selles kontekstis hindasime töös kirjeldatud andmekaeve meetodite mõju soovituste täpsusele. Doktoritöös kirjeldatud hierarhiliste regioonide põhine andmete rikastamise meetod omas selget positiivset mõju soovitajate täpsusele, võrreldes varasemas kirjanduses kasutatud tehnikatega mis ei kirjelda kohtade ruumilist ulatust ja seoseid. Meetod suurendab ka õigesti kirjeldatud kasutajate külastuste hulka, kuna suudab kirjeldada külastusi suurematesse aladesse, nagu pargid, linnajaod jne. Tulemuste põhjal võib järeldada, et linnaruumi kirjeldamine hierarhiliste regioonidena annab kasutajate tegevuste modelleerimisel olulist lisaväärtust.

Katsete käigus ei tuvastanud me olulist positiivset efekti töös kirjeldatud mürafiltri kasutamise korral. See viitab, et müra filtreerimine nõuab praktiliste rakenduste jaoks täiendavat edasiarendust.

Appendix 1

Publication I

T. Tammet, A. Luberg, and P. Järv. Sightsmap: Crowd-sourced popularity of the world places. In L. Cantoni and Z. P. Xiang, editors, *Information and Communication Technologies in Tourism 2013*, pages 314–325. Springer Berlin Heidelberg, 2013

Sightsmap: crowd-sourced popularity of the world places

Tanel Tammet[a,b], Ago Luberg[a,b], Priit Järv[a,b]

[a] Eliko Competence Centre
Tallinn, Estonia
ago.luberg@eliko.ee

[b] Tallinn University of Technology
Tallinn, Estonia
tanel.tammet@ttu.ee
priit@cc.ttu.ee

Abstract

We analyse and combine a number of world-wide crowd-sourced geotagged databases with the goal to locate, describe and rate potential tourism targets in any area in the world. In particular, we address the problem of finding representative names and top POIs for popular areas, with the main focus on sightseeing. The results are demonstrated on the sightsmap.com site presenting a zoomable and pannable tourism popularity heat map along with popularity-sorted POI markers for concrete objects.

Keywords: crowd-sourced mapping; popularity analysis; heat map; entity disambiguation

1 Introduction

The goal of this work is to build a world-wide database of the sightseeing popularity of concrete places (POI-s) and wider areas in the world, using purely crowd-sourced data. By sightseeing popularity we mean the estimate of number of people visiting the place and considering it as an interesting place for sightseeing, as opposed to very popular places with no or very little potential for sightseeing, like hospitals, schools, gas stations, bus stops and airports.

Obviously, some of the abovementioned popular non-sightseeing places like schools and railroad stations may in some exceptional cases be sightseeing places as well: famous old colleges, Grand Central Terminal of New York, etc. Two separate extremely important categories of objects in tourism industry – hotels and restaurants – are ambivalent as well: on one hand, utilitarian and not necessarily a target or cause for travelling, on the other hand, an important source of emotions and sometimes also an important partial motivation for travel.

As said, our work is focused on popular sightseeing places regardless of their category. Hence we are not using any data sources like TripAdvisor (http://www.tripadvisor.com/), Expedia (http://www.expedia.com/), UrbanSpoon (http://www.urbanspoon.com) or Zagat (http://www.zagat.com) which are primarily focused on specific categories, typically hotels and/or restaurants. Clearly, the hotels and restaurants are among the best crowd-described, -mapped, -reviewed and -rated tourism objects already.

Fig 1. A screenshot of the heat map for most of the world on a single picture, with 10 top spots (1. New York, 2. Rome, 3. Barcelona, 4. Paris, 5. Istanbul) marked. Europe, especially the belt from Netherlands to Italy as well as the mountainous areas and the Spanish coastal areas dominate. In U.S. the mountainous areas in Utah and Colorado are well marked, in addition to coastal cities. The original picture is colour-coded as a proper heat map.

The sightseeing popularity database we build is used in the sightsmap.com site for showing a zoomable and pannable touristic popularity heat map for any area in the world as an overlay on the standard Google maps (http://maps.google.com/). Popular areas on the map will be labelled with an appropriate crowd-sourced name. Concrete popular places will be also shown on the map with colour-coded markers in the order of the relative popularity in the currently visible map area.

There are numerous application possibilities for such a database. First, it is already used for showing map overlays geared towards finding interesting POI-s to visit in any region, large or small, in a uniform manner anywhere in the world. Second, the database can be used as an input for a tourism recommender like Sightsplanner (Luberg et al., 2011; Luberg et al., 2012). Third, the database can be used for doing popularity analyses for the tourism industry.

There are also several advantages to using crowd sources as contrasted to POI databases and guides already created by experts in the tourism business. The crowd-sourced approach guarantees that there are no significant holes, i.e. interesting places and areas unmarked, and that the popularity estimates are, despite inevitable

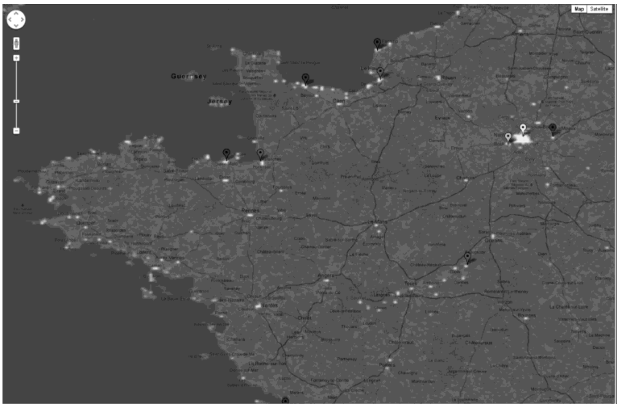

Fig 2. A screenshot of the heat map for the north-western France, with 10 top spots (1. Paris. 2. Versailles, 3. Euro Disneyland, 4. Mont Saint Michel, 5. Honfleur) marked. The castles of the Loire Valley form the central belt. The original picture is color-coded as a proper heat map.

Fig 3. A screenshot of the heat map for Manhattan, with 10 top spots marked. The open marker popup window links to the Wikipedia and Foursquare pages of the Metropolitan Museum of Art. The original picture is color-coded as a proper heat map.

fluctuations, relatively objective, which is very hard to achieve by a small number of experts. Last not least, the popularity measurements can be done uniformly and comparably all over the world.

In the next section we will provide a brief overview of the data sources and the main algorithms employed in our system. In the section 3 we will describe the relations between the data sources and the aspects of merging and enriching data in more detail. Section 4 will present experimental results and we will end our paper with related work and conclusion.

2 Different kinds of popularity and data sources

Although our methods focus on detecting sightseeing popularity, the notion is ambiguous and contains several different subcomponents (visual beauty/interestingness, general public awareness about the place, the number of actual physical visitors etc.). Each of the data sources used covers some components much better than the others; hence they complement each other well. The data sources have been harvested using their public web API-s (Panoramio (www.panoramio.com/) and Foursquare (https://foursquare.com/)) or downloaded in the already converted semantic format (Wikipedia (http://en.wikipedia.org) downloaded in the form of DBpedia RDF database, later complemented with the public Wikipedia logfiles). Harvesting and downloading has been performed during 2012.

- Our main data source Panoramio.com represents the *visual component* of sightseeing: something beautiful or interesting to see. Panoramio contains ca 44 million geotagged photos uploaded by users. For several reasons, the Panoramio photos are dominated by these with touristic and sightseeing interest (in contrast to more private photos on Flickr (http://www.flickr.com/)). Google maps and Google earth (http://www.google.com/earth/) use the Panoramio photos as their photo layer. We have downloaded only the metadata (location, photographer, title), not the actual photo files.

- The second data source Wikipedia represents the *general public awareness* about the place. We could safely say that all the interesting places, historic events, people etc. with public interest above a certain threshold do have a Wikipedia article. Places and historic events are normally geotagged in Wikipedia. The popularity – the exact number of readings in a selected time period – of each Wikipedia article can be obtained from the publicly available logfiles. We are using ca 700 000 geotagged Wikipedia articles with types which do not indicate noninterestingness for touristic purposes (like articles about plants, animals, people). We use full logfiles for two days, one selected from summer, the other from winter.

- The third data source Wikitravel (http://wikitravel.org/) essentially complements Wikipedia: places: above a certain touristic interestingness threshold normally have a Wikitravel article corresponding to some Wikipedia article. We are using the list of existing Wikitravel article names to detect whether a Wikipedia article has a complementing Wikitravel article as well.

- The fourth data source Foursquare gives an estimate of the *number of people actually visiting the place*. A large percentage of visits (and a large percentage of Foursquare places) are done and created by local people visiting offices and eating lunch. Foursquare, differently from all the above sources, has a fairly detailed and well-used system for the crowd-sourced typing of places. We have downloaded not the whole Foursquare places database, but only ca 2 000 000 places, taking the places Foursquare presents when asked for a circle around some of the top hotspots we have previously found out from the analysis of the Sightsmap photos. We harvest several concentric circles around each place previously determined to be visually popular enough: small circles for objects in the cities and large circles outside or around the cities. In the other words, we have only downloaded the more popular Foursquare places in the neighbourhood of the more visually popular (world-wide) places.

3 Heat map generation, basic labelling and data merging

The heat map generation has two separate outcomes. First, it generates the visual heat map overlays for the map. We use the browser-based Google maps as the underlying map. Second, it generates a detailed popularity data for each small rectangular area (a pixel on the heat map) for each zoom level, which is later used for labelling, harvesting additional information etc.

The heat map generation is done separately for six different zoom levels of the world, each with each own granularity. Additionally, the seventh layer is a set of high-resolution heat maps, each typically covering one city, created for ca 15000 top spots in the world. The resolution of these high-resolution heat maps depends on the popularity rank of the hotspots: the more photos, the higher the resolution, up to the street level for the top 500.

Our algorithm takes into account both the number of photos and the number of separate photographers in the Panoramio database for each area. The colour of each pixel on the heat map is calculated by a logarithm-like root function, different for each zoom layer. We use one byte for the colour information, with the the top popular places being bright yellow, followed by orange, red, purple and blue hues.

3.1 Basic labelling with Wikipedia

The pure visual popularity heat map lacks a clear indication of what exactly is there in a hot area. In short, the top spots in each view have to be marked and the markers should ideally contain the name and the pointers to the most relevant information about the places.

Our basic solution for creating these markers, finding the titles and providing pointers is to look for a most popular geotagged Wikipedia article at or very close to each top hotspot at each heat map grid. Articles with an obviously unsuitable type (like plants, animals, and people) are excluded. This method guarantees that, for example, on the whole-world view where each hotspot pixel corresponds to a relatively large area, we automatically get the Wikipedia city articles as the most popular, but as we zoom in, the area for each pixel becomes smaller and we will start getting markers and articles about villages, beaches, castles etc.

The actual algorithm is the following. First we cluster the heat map dots to avoid showing lots of markers very close to each other. Then we look for the most popular Wikipedia articles near the hotspots: the higher-ranked a heat map spot is, the larger the area to search. If nothing is found or the found article has a much lower popularity than the heat map spot, we do not attach anything to the hotspot. Otherwise we connect a hotspot to the Wikipedia article plus the corresponding Wikitravel article, if available.

As mentioned before, in order to generate the popularity data and a popularity-sorted list of Wikipedia articles we use the logfiles mentioned before plus an additional coefficient giving a significant bonus to Wikipedia articles with a type suitable for sightseeing, for example, world heritage sites.

It is worth noting that knowing a highest-ranked Wikipedia article for an area helps users to google for more, since the article always gives us a title of the place to look for.

3.2 Basic merging with Foursquare

The ultra-high-res heat maps for which we do load Foursquare data is populated with the combined Wikipedia and Foursquare markers for top spots in the heat map, using an algorithm which – similarly to the Wikipedia labelling algorithm from the previous chapter – first tries to associate Wikipedia and Foursquare objects to the most popular places on the map and finally interleaves the remaining, unmatched top Wikipedia and Foursquare articles to the mix, even if they are not located near a visually attractive spot.

Foursquare places merging with Wikipedia articles is performed using an algorithm which takes into account both the geographical distance and a similarity of the names of the place vs. the article. In order to be merged, both of these parameters must be sufficiently similar.

Foursquare locations are ordered based on the combination of different users ever checked in and the type of the place. First, we exclude both geotagged Wikipedia articles and Foursquare locations with obviously non-geographic or non-sightseeing type (homes, offices, bus stops etc.). Second, we add bonuses to articles and locations based on the suitability of their type: for example, castles, churches and public squares get different bonuses.

In most cases the geographical coordinates of the underlying visually popular spot, the closest popular Wikipedia article and the corresponding Foursquare location (close both by coordinates and the name), as well as the name of the article/location are noticeably different. We use a relatively complex heuristic algorithm to determine the most suitable name and coordinate to present for the user as a marker. The percentage of errors our algorithm makes varies a lot for different zoom levels and regions and has not been measured with a sufficient quality to present it in the paper.

4 Labelling areas and merging objects: issues and improvements

The general idea behind labelling visual hotspots was briefly described above. Here we will present some main problems we have encountered and propose ways to improve our system.

For every visual hotspot we try to to find a matching Wikipedia article. A significant percentage of popular hotspots will get a match from Wikipedia. We try to find the name for non-matching objects by looking at Panoramio pictures nearby. We take a certain area around the hotspot (for example, 1 km radius) and look at the titles of pictures within that area. Based on this information we try to get the name of the object in the hotspot.

Table 1. An example of candidate list for pictures near Cliffs of Moher. The best match is has rank 1 and *n* 3 (marked with italics). Some less frequent candidates are omitted.

Candidate	n	Rank	Pos	Total	%
moher	1	1	656	859	76.4
of	1	2	631	859	73.5
cliffs	1	3	587	859	68.3
of moher	2	1	595	859	69.3
cliffs of	2	2	559	859	65.1
moher ireland	2	3	67	859	7.8
cliffs of moher	*3*	*1*	*534*	*859*	*62.2*
of moher ireland	3	2	64	859	7.5
cliffs of moher ireland	4	1	60	859	7.0

The title of the picture is tokenised into lower case words. We ignore commas, full-stops etc. For every tokenised title we will find the word *n-grams* for *n* being from 1 to 4. An *n-gram* is combined by taking *n* consecutive words from the title. A simple example: given a title "A picture of Big Ben", we will end up with tokens: "a", "picture", "of", "big", "ben". All *1-grams* are: "a", "picture", "of", "big", "ben". And all *4-grams* are: "a picture of big", "picture of big ben".

After finding *n-grams* for every picture in the area of interest, we take the 5 most frequent *n-grams* for every *n*. We will end up having up to 20 *n-grams* (5 most frequent for every *n=1..4)* for a hotspot which we consider name candidates.

An example candidate list for "Cliffs of Moher" (pictures near Lahinch, Galway in Ireland) is presented in Table 1. The column *n* stands for *n* used in *n-gram* (how many tokens is used to form up a candidate), *Rank* stands for rank in current *n* (1 being the most frequent *n-gram*), *Pos* ("positive" pictures) is a number of pictures which contain the given *n-gram*, *Total* represents the total number of pictures near by and *%* shows the percentage of "positive" pictures. We have marked the correct candidate in the table.

The given example illustrates already some problems we have with this methodology. After generating a list of candidates, we have to pick the correct candidate. Finding the correct one is not so straightforward. It is obvious that we cannot use the most frequent candidate as the final name, because it may-be just part of our final name. If our final name consists of 3 words, then every word alone in this name has at least the same or even higher frequency. This is very clear in the example: “cliffs”, “of” and “moher” all have higher frequency than “cliffs of moher” together.

The idea we have with the candidate selection is to find the longest candidate which has frequency above a certain threshold. For example, if the threshold is 30%, then we would find “cliffs of moher” to be the best candidate. To improve the precision, we are planning to apply machine learning to find the best threshold (or may-be even have additional indicators for the best pick in addition to frequency and term count).

Another problem is more related to the concept of taking pictures. It often happens that bigger (high) objects can be captured only from distance. It is very hard to take a picture of Eiffel Tower when being right in front of it. The same applies for our example “Cliffs of Moher”. The candidate list we presented earlier is actually taken from about 2 kilometres from the object itself (object location based on Wikipedia). Wikipedia location for the cliffs has about 400 pictures and 267 mention “Cliffs of Moher”, while 2 kilometres away the count of pictures is about 800 and 534 of those mention the correct object.

For our system, we actually need both those places. If later we want to have a recommendation of the best sightseeing places, we can prefer the distant location to take pictures. The 2 kilometre gap between the objects makes it harder to merge them into one. Currently we will have two separate objects (even though the name of two places could be the same).

In the next section we will present some experiments with Panoramio picture titles. All the work presented is based on the methodology described in the current section.

5 Experiments and Results

We use two different datasets for our tests: pictures from United Kingdom and pictures from France. For every popular place we have found up to 20 possible candidate titles. In order to evaluate our simple approach, we use Wikipedia to extract titles of popular objects. For every popular object we find a Wikipedia article with the same or close geocoordinates. In case there are several Wikipedia pages for one location, we try to take the most appropriate (popular and type-wise suitable). Obviously, not all visually popular locations have a Wikipedia entry. In our evaluation we only consider those locations which have a linked Wikipedia article. After generating all the *n-gram* candidates for a location we will see whether the Wikipedia name is within those candidates. Statistics about the datasets can be found in Table 2.

Table 2. Statistics about the datasets for UK and France.

Property	**UK**	**France**
Hotspots	14 768	13 621
Wikipedia objects	9753	9931
Panoramio picture count	1.4M	1.5M
Wikipedia object match	5458	5531
Match %	56%	56%

As shown in the Table 2, we were able to find about 56% Wikipedia objects from the Panoramio pictures. This means that the Wikipedia name matches (we allowed *Levenshtein distance* (Levenshtein distance, edit distance, http://en.wikipedia.org/wiki/Levenshtein_distance) up to 3) with one candidate. We outline several reasons why some objects are not found/matched:

- The number of pictures in the close vicinity is very low (or even zero). If we have an object and only 3 pictures mention that object, we want to look at pictures from the bigger area. We can extend the search area, and end up with 20 new pictures, but none of those mention the object we were looking for (all the new pictures mention some other object).

- Wikipedia and Panoramio coordinates do not match. We look only those matches which are close to each other. For our matching evaluation we need Wikipedia and Panoramio pictures to be very close. It may happen that the source data has somewhat rounded coordinates (0.01 difference in latitude or longitude number can mean 1 km distance). Another possibility is that some objects are usually pictured from a distance. A good example was given in the previous section about Cliffs of Moher.

- Different name variants. In Wikipedia, some objects have additional information like county or country in their titles. For the Wikipedia place

"Lincoln, England" we have found an n-gram "Lincoln", which is a correct match. These kinds of matches are not counted in our "match" number.

- The Panoramio title is too general. For some objects, there are a lot of pictures which indicate the name of the city or county where the object is located. For example, the case where there are 100 pictures near a certain Wikipedia object and only 3 mention the object itself. Other pictures mention the city, the county etc. It can easily happen that more general n-grams push the correct object out.

Our dataset for the described experiments has about 14 000 "hotspot" objects and about 10 000 Wikipedia objects. For the objects with Wikipedia articles, we could combine Panoramio and Wikipedia data to validate the title of the object. For the rest, we have to rely on Panoramio pictures (or on some additional external data source). Usually the title generated from the Panoramio title *n-grams* is not wrong, but it might be too general or a slightly different variation than Wikipedia article would have. We estimate that the Panoramio based object titles are correct in at least 56% of cases. If we add different name variations and more general objects, we might end up with 70-80 %.

6 Related Work

Heatmaps are used in various domains in order to visualise intensity of a certain values. We mention few which are also related to tourism. Fisher (2007) uses tile download statistics from Microsoft map server to present popular areas. He calls the system Hotmap. Every time a user looks a map, she downloads visible tiles from the server. Objects (and tiles) which are watched more often, have higher download numbers and they will become more popular for Hotmap. They present different ways to use heatmaps mentioning also a possibility to draw users' attention to prominent objects.

Kurata (2012) presents a potential-of-interest map based on Flickr pictures in Yokohoma. He present an interesting approach for finding popularity of objects where only pictures from non-local users are taken into account. Users who live in the city, are considered as non-tourists and their pictures do not add popularity. In our case, to find the name and the type of the object, we have to use pictures from local people. And it may happen, that those are even more accurate than tourist pictures, as a tourist may not know the exact name of the object. Kurata presents user evaluation which is very valuable and something we still have to organise for our recommender system.

Crandall et al. (2009) describe their system which uses image textual and visual features to group pictures into popular objects. They find a name and a descriptive picture for every popular object. Processing image textual information is very close to what we have presented in our paper. They use distinctiveness to order name candidates instead of using candidate name ratio to all pictures near-by the object. They present a machine learning technique usage for solving the problem of naming the objects (where the photo is taken). Although they present that combining textual and visual features yield the best results, we keep our focus on using only textual information.

Alves et al. (2009) present KUSCO system which deals with enriching POI data. They extract information from search engine to gather web pages about a certain POI. Then they use natural language processing to extract concepts for objects. An interesting idea is to use WordNet (http://wordnet.princeton.edu/) concepts matched with words from the web pages. We have started working on something similar: we try to extract words from Panoramio picture titles and find similarity or distance between found words and WordNet concepts. We only consider certain concepts from WordNet which represent categories of POI: museum, restaurant, hotel, church etc.

Popescu et al. (2008) present a system which integrates Wikipedia and Panoramio in order to identify geographical names, categorise objects, find geographical coordinates and rank objects. They use Panoramio picture count as one possible rank for objects (more pictures means higher rank). They also try to find categories for objects where they use language processing from the first sentence of Wikipedia and web search. They compare their system with Geonames (http://www.geonames.org/), but they do not use Geonames as a source for their data. A lot of our ideas align with their proposed solutions: using Panoramio for ranking objects, merge objects with Wikipedia, try detecting categories from web search (something we are currently working on).

Popescu et al. (2009) present a multilingual geographical gazetteer creation based on Flickr, Panoramio, Wikipedia and web search. They detect place names using a vocabulary with geographical concepts. They also present object ranking and categorisation. They have improved some of the methods compared to their paper from 2008. They use Flickr instead of Panoramio. They have also published their gazetteer which can be downloaded (http://georama-project.labs.exalead.com/gazetiki.htm). We could evaluate our system against this dataset. However, we need to implement some additional functionality before doing the evaluation, in order to perform full range comparison.

Zheng et al. (2009) describe a system for building a world-wide landmark database. They use pictures from Picasa (http://picasa.google.com) and Panoramio along with Google Image Search (http://images.google.com) to download picture files. They also use textual information from Wikitravel to complement objects which are not present in pictures. They use picture and Wikitravel text information to find the name for the popular object. Image processing helps to detect different pictures about the same object which can be clustered into one group. In addition to image processing they use picture title word n-grams – the most frequent n-gram is used as the title for the group.

7 Conclusions and Future Work

We have presented the Sightsmap system with a goal to build a world-wide database of the sightseeing popularity of concrete POI-s. We are using purely crowd-sourced data: Panoramio, Wikipedia, Wikitravel, Foursquare. While the main goal is to detect popularity, first we have to tackle different data extraction and integration problems. We have presented experiments on finding an object name from the Panoramio picture titles. We have also described the way to gather information and to use different sources to calculate popularity for objects in the world. We have presented a heat map solution sightsmap.com, where all our data is put to use.

One of the future plans is to be able to recommend objects all over the world. The recommendation should be based on the interests of the tourist, hence we need to find a category for every object in the world. We have already started working on this goal and have briefly mentioned our ideas on the subject.

References

Alves, A., Pereira, F., Biderman, A. & Ratti, C. (2009). Place Enrichment by Mining the Web. In M. Tscheligi, B. de Ruyter et al. (Eds.), *Ambient Intelligence*, 5859: 66-77. Springer Berlin.

Crandall, D. J., Backstrom, L., Huttenlocher, D., & Kleinberg, J. (2009). Mapping the world's photos. In *Proceedings of the 18th international conference on World wide web:* 761-770. ACM.

Fisher, D. (2007). Hotmap: Looking at Geographic Attention. *IEEE Transactions on Visualization and Computer Graphics 13*(6): 1184-1191.

Kurata, Y. (20120) Potential-of-Interest Maps for Mobile Tourist Information Services. *Information and Communication Technologies in Tourist 2012:* 239-248. Springer Vienna.

Luberg, A., Järv, P., Tammet, T. (2012) Information extraction for a tourist recommender system. *Information and Communication Technologies in Tourism 2012:* 343-354. Springer Vienna.

Luberg, A., Tammet, T. & Järv, P. (2011). Smart City: A Rule-based Tourist Recommendation. *Information and Communication Technologies in Tourism 2011:* 51-63. Springer New York.

Popescu, A., Grefenstette, G., Bouamor, H. (2009). Mining a Multilingual Geographical Gazetteer from the Web. *Proceedings of the 2009 IEEE/WIC/ACM International Joint Conference on Web Intelligence and Intelligent Agent Technology* (WI-IAT '09), Vol. 1: 58-65. IEEE Washington.

Popescu, A., Grefenstette, G., Moëllic, P.-A. (2008). Gazetiki: automatic creation of a geographical gazetteer. *Proceedings of the 8th ACM/IEEE-CS joint conference on Digital libraries* (JCDL '08): 85-93. ACM New York.

Zheng, Y. T., Zhao, M., Song, Y., Adam, H., Buddemeier, U., Bissacco, A., ... & Neven, H. (2009, June). Tour the world: building a web-scale landmark recognition engine. In *Computer Vision and Pattern Recognition*, 2009: 1085-1092.

Acknowledgements

This research has been supported by European Regional Development Fund.

Appendix 2

Publication II

P. Järv. Extracting human mobility data from geo-tagged photos. In *Proceedings of the PredictGIS'17:1st ACM SIGSPATIAL Workshop on Prediction of Human Mobility*, pages 4:1–4:7, 2017

Extracting Human Mobility Data from Geo-tagged Photos

Priit Järv
Tallinn University of Technology, School of Information Technologies
Akadeemia tee 15
Tallinn, Estonia 12618
priit@whitedb.org

ABSTRACT

Photos shared by users on public websites (Flickr, Panoramio) provide a resource for mining behavioural data. When the photos are associated with locations and time stamps, we can reconstruct the trajectories of the users and use the resulting mobility traces for learning behaviour patterns. In this paper we focus on two aspects of mobility traces: noise filtering and semantic annotation.

The extracted trajectories are initially noisy due to errors in geographical coordinates and time stamps. We show how such noise can be partially filtered and evaluate the performance of the filtering on a synthetic dataset.

To make use of the mobility traces, an essential step is semantic annotation. Places or activities are associated with segments of the traces. This is frequently performed by integrating a database of relevant places and associating them by proximity. We demonstrate that the popularity of the places, if available, can improve the association accuracy. In our experiment, the accuracy of automatic annotation increases from 60% to 68%.

CCS CONCEPTS

• **Information systems** → **Data extraction and integration**; *Recommender systems*; *Clustering and classification*;

KEYWORDS

Geo-tagged photos, Human mobility, Noise filtering, Semantic trajectories

ACM Reference format:
Priit Järv. 2017. Extracting Human Mobility Data from Geo-tagged Photos. In *Proceedings of PredictGIS'17:1st ACM SIGSPATIAL Workshop on Prediction of Human Mobility , Redondo Beach, CA, USA, November 7–10, 2017 (PredictGIS'17),* 7 pages.
https://doi.org/10.1145/3152341.3152346

PredictGIS'17, November 7–10, 2017, Redondo Beach, CA, USA

ACM ISBN 978-1-4503-5501-8/17/11. . . $15.00
https://doi.org/10.1145/3152341.3152346

1 INTRODUCTION

Public geo-tagged photos from sites like Flickr and (until 2016) Panoramio provide a readily available source for human mobility data. We can learn behaviour patterns by taking the locations and time stamps of individual photos and reconstructing the movement trajectory of the photographer. Such trajectories are called mobility traces. To make them more useful, mobility traces are semantically annotated by adding the names of visited places or performed activities.

Geo-tagged photos as a source of mobility traces have distinct characteristics when compared to GPS and cell phone traces. People generally take photos in interesting locations. This helps in applications like tourism recommendation, making each individual data point more significant and compensating for the sparsity of data.

Time stamps and coordinates of photos do not always represent the time and place the photo was taken. Either human error or the software workflow from the camera to the photo sharing website can make the metadata of the photo inaccurate. To truthfully represent the movement of the person, such noise should be removed from mobility traces.

We solve this problem by finding "noisy" photos or groups of photos. Noise can be identified by sudden high speed "jumps" from one place to another, that would be unlikely by usual means of sightseeing on foot or in a vehicle. We determine heuristically which photos cause the speed anomaly and remove them. The remainder is a "clean" mobility trace.

A common challenge in working with mobility traces is semantic annotation. To extract common patterns of activities, the trace that is initially a sequence of coordinates is enriched with annotations like the movement of the person ("stopping", "moving"), interpretation of the activity ("work", "home" and "lunch") or the name of the place the person visited.

In this paper we focus on adding place semantics. A common method is to integrate points of interest (POIs) from a pre-existing database. If there are POIs within a given radius from a point being annotated, the closest POI is chosen. If there are no such POIs, the point is left without annotation. The drawback of this method is that whenever there is a choice of multiple nearby places, there is the possibility that some of them attract much more visitors than the others.

When the data about the popularity of places is available, it can be used to improve the annotation accuracy in such cases. We describe a method of combining popularity with proximity that borrows from the concept of gravity in physics. More popular places extend their influence at a spatially wider area. We give a simple probabilistic interpretation of the popularity that allows scaling it properly for the calculation. We experiment with this concept using a manually verified ground truth dataset of photos.

The rest of the paper is organized as follows. In Section 2 we review related work. In Section 3 we describe the process of creating semantically annotated traces from individual photos. Section 4 discusses the experiments made with synthetic and real-world Panoramio datasets. In Section 5 we summarize the results and outline the direction of future work.

2 RELATED WORK

Human mobility data can be harvested from various public sources. Li et al. extract trajectories from EXIF metadata of photos and coordinate/username records from Panoramio [16]. Girardin et al. analyse EXIF information and annotations from Flickr to extract tourist flows [11]. Recommender systems that learn mobility patterns from geo-tagged photos have been designed using Panoramio [20] and Flickr [4, 7, 15, 19]. Location based social networks (LBSN) check-ins and geo-located tweets in Twitter are less frequently used to mine trajectories, but a large body of research exists on exploiting them for behavioural pattern analysis [22, 25]. Non-public, but frequently used sources of human mobility include GPS traces [29] and data collected by mobile phone operators [14].

Choudhury et al. outline the basic procedure of extracting mobility traces where the waypoints are individual photos. They identify the subset of photos taken within the subject area and the subset of users that matches their application of itinerary recommendation. They arrange the photos into streams ordered by time stamps. The streams are split at large gaps into daily tours. [7]

Alternatively, the extraction process may be integrated with identifying important locations. Lu et al. cluster the photos by their location before extracting mobility traces. Paths that connect clusters, representing the densest photographed locations, are then extracted [20]. Brilhante et al. first build a database of points of interest (POIs) in the region. They associate photos with POIs by proximity, creating paths that connect POIs [4].

Noise filtering in photo-derived mobility traces has been addressed in specific cases. Invalid time stamps may be detected in Flickr photos by comparing them to the upload dates [7]. Lim et al. use prior information about geolocation accuracy [19]. Other social networks may be affected by different sources of noise, for example Zhang et al. found that 75% of Foursquare check-ins were forged in their study, to receive in-system rewards [30].

For trajectories in general, there are three main noise filtering methods: detecting noise by absolute speed threshold or smoothing by statistical (median filter) or physical (Kalman filter) modelling [31]. Smoothing techniques are not appropriate for photo-based traces, because relocating individual trajectory points would change their semantics. In this paper, we use the speed threshold heuristic adapted to sparse traces.

Semantic annotation adds labels to trajectory waypoints or segments that assist in utilizing the trajectories. Some type of annotations are possible using only the trajectory data. GPS trajectories can be segmented to identify locations of interest ("stay points" or "stops") [17, 23]. Liao et al. annotate GPS traces with activities using a probabilistic model called conditional random fields [18].

Place semantics require integrating additional sources. A common method is to use a database of POIs and associate the nearest POI within a given radius to trace segments or trace points [4, 7]. Furletti et al. present a gravity model where the association probability is proportional to the relevance of POIs and inversely proportional to their squared distance. They use the method to infer activities instead of associating individual POIs [9].

A promising approach to adding place semantics is using regions of interest (ROI) that represent the spatial extent of places [10, 28]. Chen et al. review different methods and apply a technique based on stay points, dense areas of trace points that span a sufficient interval in time [6].

Semantic annotation has wider applications and perspectives. For example, Skoumas et al. mine geographic and semantic information from user generated texts using natural language processing [26]. They apply the semantic annotation to paths in a routing application. Arase et al. classify and label entire traces by theme. They use spatial, temporal, behavioural and textual features [2]. Andrienko et al. deal with a wide range of place semantics issues, such as privacy and scalability. They provide a method based on temporal visit patterns in trajectories. However, their visual-aided method is interactive. The authors state that it is not viable to solve the problem fully automatically [1]. A review of semantic trajectories topics is given in [24].

Semantic trajectories and human mobility are part of a wider subject defined as urban computing [32]. A more comprehensive overview of technical aspects in mobility traces can be found in trajectory data mining reviews [21, 31].

3 METHODS

3.1 Data acquisition and preparation

We will briefly outline the tasks in the acquisition and preparation stage, as the details depend on the application. The mobility data is collected by downloading the metadata of the photos: coordinates, time stamps and user identifiers. To extract the mobility of individuals, the photos are grouped by users.

Our approach works on a sequence of photos representing a contiguous tour of a single user of a photo sharing website during a single day. Such sequences can be extracted by sorting photos by time stamps and partitioning it at places where the user is presumed to be sleeping or resting. This can be determined either by length of time gaps or by absolute values of time stamps.

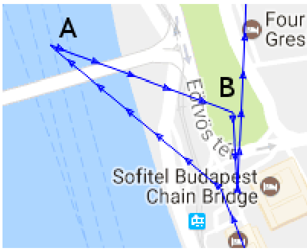

Figure 1: Correctable noise in a trajectory constructed with photo coordinates and time stamps. The photographs A and B were taken only 26 seconds apart, resulting in relatively high estimated travel speed from A to B. Removing A results in a smooth trajectory. Map provided by Google.

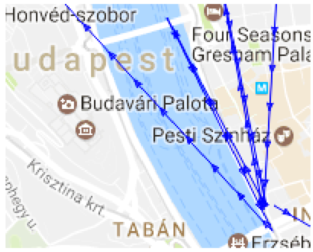

Figure 2: Time stamp noise. The erratic trajectory suggests that the original visit times have not been preserved. Map provided by Google.

3.2 Noise filtering

Three factors contribute to the noisiness of mobility traces. The coordinates of photos are sometimes altered by the user to represent the actual location of the subject of the photograph, if it is taken from a distance. In addition, when the camera is not equipped with a positioning device, the users may misidentify photo subjects. Finally, another kind of noise is introduced if the time stamps in image metadata are changed.

Our filtering method is suited to address the noisiness of the position of single photographs (Figure 1). The incorrect time stamps can result in a very erratic trajectory and make it unfeasible to reconstruct the movement of the user (Figure 2).

The filter works by calculating the speed of transition between consecutive points in the mobility trace. If the speed is too high, one of the points is likely to be "noisy" and should be removed. Low extremes of speed are always acceptable, as they represent staying at a location.

We calculate realistic distances between photos using OpenStreetMap (https://www.openstreetmap.org) and the Open Source Routing Machine (OSRM) router (http://project-osrm.org/). For distances below 30m we use straight line distance. When calculating speeds, we must also consider that for photos taken in close proximity intervals in time stamps can be small or even 0. We desensitize the filter to such micro-movements by compensating very small intervals when the distances are also small to reduce the variance of speed.

The filter has two parameters. The cut-off speed v_{cut} determines, which transitions between consecutive photos are considered "noisy". The noise ratio threshold r_{noise} sets the maximum allowed ratio of noise per single trace.

When the estimated transition speed $v_{i,i+1}$ between two photos i and $i+1$ is above the cut-off v_{cut}, we use the following heuristic to remove one of the photos:

- If i is the first photo in the trace, compare $v_{i,i+2}$ and $v_{i+1,i+2}$. Whichever of these is smaller corresponds to a better candidate as the first photo. Note that either of the speeds may still be above the threshold and subsequently cause another removal.
- If $i+1$ is the last photo in the trace, compare $v_{i-1,i+1}$ and $v_{i-1,i}$ to determine, whether i or $i+1$ will be a better candidate for the last photo in the trace. Again we prefer the lower speed.
- In other cases, determine which removal would result in a lower speed trajectory by comparing $v_{i-1,i+1}$ and $v_{i,i+2}$.

For each trace we calculate the ratio of the number of photos cut n_{cut} to the number of photos in the trace n: $r_{cut} = \frac{n_{cut}}{n}$. If $r_{cut} \geq r_{noise}$, the trace is considered unreliable and discarded. The high amount of noise may be an indication of incorrect time stamps. The removals also make it less likely that the original visit sequence of places is preserved after filtering.

3.3 Place semantics

Before associating points of interest (POIs) with the trace, the individual photos are clustered by proximity to each other. This partitions the trace into "stop" and "move" episodes, although we do not add these annotations explicitly. Places are associated with "stops", clusters of photos. Figure 3 illustrates the result of semantic annotation, with photos on the trajectory being clustered and the cluster associated with a nearby POI (marker with the dot). For the purpose of finding candidate POIs we assign a radius to each cluster that represents how spread out are the photos that form the cluster.

Even a single photo potentially represents a legitimate visit of a place. Therefore, the role of the clustering is only to group adjacent photos. Because we are dealing with trajectories, it is only meaningful to cluster photos that directly follow each other in the trace timeline.

Using the above observations, clustering can be simplified by converting the data into one-dimensional space: we place the start of the trace at coordinate 0 and each subsequent point i at a cumulatively increasing distance $d_i = d_{i-1} + \delta_{i-1,i}$ where $\delta_{i-1,i}$ represents the actual travel distance between two points calculated using OSRM. After the space conversion,

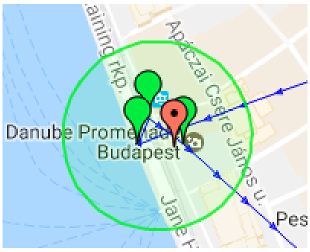

Figure 3: A cluster of photos on a trajectory associated with a POI (the dotted marker). Bank of Danube, Budapest. Map provided by Google.

clustering can be performed with a standard density-based method and a standard Euclidean distance metric.

We associate POIs with clusters by the formula

$$\arg\max_{l \in L} \frac{popularity_l}{r_l^2}$$

where l is a POI in the set of locations L that fall within the cluster radius and r_l is the distance between the POI and the centroid of the cluster. In areas with high POI density, this tends to associate the cluster with the most relevant POI close to its centroid. If we set $popularity_l = 1$ for all POIs, the method degenerates to proximity based association.

The "attraction" of the POI decreases proportionally to the square of the distance, but some POIs initially have higher attraction. We propose the following geometric interpretation to assist in normalizing and scaling $popularity_l$. Assume that l_1 is visited 4 times more frequently than l_2. If we draw circles on the map representing the places, with the area of l_1 4 times larger and select a point at random, the random point is 4 times more likely to be inside l_1-s circle. Therefore, $popularity_l$ should be proportional to the likelihood that l is visited by a user.

To normalize popularities, we assign $popularity_l = 1$ to the least popular places (or those for which there is no data available). A more popular place should take precedence over a less popular one if they are equally distanced. As the distance to the more popular place increases, at some point the less popular place should be chosen, as it is much closer. We then consider the distances at which the most popular places should dominate least popular places. For example, if the unpopular place is at a distance of 50m and we prefer that the popular place should be chosen instead, if it is up to 200m away, the maximum value of $popularity_{l_{max}} = \left(\frac{200}{50}\right)^2$.

4 RESULTS

We tested the methods with three datasets:

- A set of 177496 photos of Budapest from Panoramio.
- A subset of traces from above with manually verified place annotations.
- A synthetic dataset with simulated noise.

We separately measured the performance of the noise filtering depending on the filter parameters and the accuracy of our proposed association method, when compared to the proximity based association. The noise filter was tested on the synthetic dataset. Place association was tested against the manually verified ground truth dataset. The experiments were done with the assumption that the application of the mobility traces is to train a tourist recommender system. This affects some decisions taken in data processing.

4.1 Data preparation

We downloaded the real-world dataset of photos of Budapest through the Panoramio API. The time stamps of photos were obtained by scraping the EXIF metadata of photos from the Panoramio website. Prior to splitting to traces, we removed users that took photos within a period longer than 21 days, because they are less likely to be tourists. Users that posted only a single photo were also removed. The rest of the photos were sorted by time stamps and split into daily mobility traces, with splitting points at gaps of 8 or more hours. Total of 4061 traces with more than one photo were produced.

We created a ground truth dataset of 100 of the longest traces by manually annotating them with visited points of interest (POIs). The visited places were identified using Google maps, the contents of the photos and user provided captions. In the few cases where the best fitting POI was still ambiguous, we consistently chose the same POI in all traces that visited the location.

To make the synthetic dataset we created clean traces that represented realistic visitor behaviour. We generated sequences of places to visit by using a 1st order Markov model trained from the ground truth dataset of 100 real-world traces. For each place, we then generated photos, with their number and placement drawn from the distributions representing the ground truth dataset. Due to the bias of selecting long traces into the ground truth dataset, the length (number of places visited) of generated traces was randomly drawn from the distribution representing the full unvalidated real world dataset.

To simulate movement between consecutive places, we used speed estimates from datasets representing real traffic. We used normal distribution with $\mu = 4.4$, $\sigma = 0.75$ for urban walking speed [5] and $\mu = 22$, $\sigma = 9.0$ for urban vehicle speed [13]. 3% of the traces were simulated as movement by vehicle, the rest as movement on foot.

We then added three types of noise to the synthetic dataset. The distribution of noise was estimated by applying the filter to a real dataset, but in several cases we have artificially increased the probability of the noise appearing. This is to counter the possible bias caused by the filter not detecting some photos.

Time stamp noise was generated by randomly overwriting time stamps in 8% of the traces. In each case the time stamps were replaced with simulated "uploading" time stamps in 5 levels of detection difficulty. The longer the interval between the time stamps, the more difficult it is to detect the noise, as resulting estimated speeds become lower.

We added new photos to 20% of the traces to simulate two types of coordinate noise, both as an isolated photo and inside a dense group of photos. The isolated photo may be caused by the user being mistaken about what the photograph depicts, or taking a photo of a remote object and modifying the coordinates so that they no longer represent the user's actual location. The noisy photo grouped together with others is due to the user adjusting the coordinates manually, usually because the photograph is of a remote object.

In the first case we added a photo with larger positional displacement ($\mu = 1km$), with the time stamp at roughly equal intervals from the previous photo and the next photo. In the second case we used smaller displacement ($\mu = 300m$), and the photograph is taken at the same time with other photos (with an interval of no more than 100 seconds).

To compute distances we used both the `foot` and `car` routing profiles of OSRM. A route was computed for both modes of movement and the shorter distance chosen.

4.2 Noise filtering

We applied the noise filter to the synthetic dataset of 9910 traces. To measure the effect of parameters, we fixed the noise ratio $r_{noise} = 0.15$ and speed cut-off $v_{cut} = 10km/h$ in turn and varied the other parameter. We then calculated the precision (how many of the removed places were real noise) and recall (how clean was the resulting dataset). In both cases, choosing the parameter involves a trade-off between precision and recall (Figure 4).

The precision of the filter is directly dependent on the distribution of estimated speeds in the dataset (left plot, grey area). When v_{cut} becomes closer to the median of the speed, precision starts dropping rapidly as the rate of false positives increases. Meanwhile, recall increases steadily. Selecting v_{cut} should then be done based on how much reduction of the original dataset is acceptable in the application.

The precision is also lowered when the accepted noise ratio threshold becomes more aggressive (right plot). This is because v_{cut} is fixed at a value that causes some false positives. As r_{noise} is lowered, more of the traces become affected, even though the number of false positives stays fixed.

The filter is more successful in detecting coordinate noise. At $v_{cut} = 10km/h$, $r_{noise} = 0.15$, 97% of the coordinate noise and 50% of the time stamp noise was detected. Even at $v_{cut} = 50km/h$, 78% of the coordinate noise was detected, while detection ratio of time stamp noise was only 21%.

Choosing the filter parameters can be guided by their physical meaning. For example, if the application focuses on pedestrian movement, $v_{cut} = 10km/h$ is quite reasonable and leaves some headroom for map routing inaccuracies. The choice of r_{noise} is dependent on how important is preserving the exact sequence of visited places in the application. r_{noise} can be relaxed if avoiding false positives is more important than preserving sequence information.

4.3 Place semantics

We measured place association accuracy by taking the 100 traces that belong to the ground truth dataset and applied clustering and automatic POI association. No filtering was applied. We measured the accuracy by comparing each image in the annotated traces with the same image in the ground truth dataset. If the association was the same POI, the annotation was considered correct. If no POI was associated automatically and the image in ground truth dataset also had no POI associated, the annotation was again correct. In other cases the annotation was considered incorrect.

For clustering we used an implementation of DENCLUE 2.0 [12] with truncated Gaussian kernel, with the bandwidth parameter $h = 30m$. We chose the truncated kernel as a simplification to ignore very weak attraction by points separated by a large distance. The threshold parameter $\tau = 1.0$ was straightforward to choose by recalling that even a single photo (with local density 1.0) is considered a cluster.

To associate the clusters with place names, we used the dataset of POIs from Sightsmap [27]. For the baseline proximity based association, the radius in which the POIs are searched is the cluster radius $r = max(100m, d_{max} + 10m)$ where d_{max} is the maximum distance between a photo in the cluster and the cluster's centroid. For the gravity association, we searched for POIs in the radius $2r$.

The popularity data for the POIs was derived by counting the number n_l of visits to each place in the ground truth dataset. We set $popularity_l = n_l + 1$, because the maximum number of visits in the dataset was 21, so no further normalization was required. For places not appearing in the ground truth dataset we assigned $popularity_l = 1$. Using this scale, the "radius of influence" of the most popular place was 4.6 times larger than that of the least popular places

The baseline proximity based method annotated 60% of the 4360 photos in 100 traces correctly. The "gravity" based, or combined proximity and popularity method, improved the accuracy to 68%.

We evaluated the effect size of the proposed method and the statistical significance of the result with McNemar's test. Table 1 shows how the chosen method affects individual associations. In 666 cases the "gravity" method changed an incorrect association into a correct one (column 2, bottom

Table 1: Estimated effect size of using combined proximity and popularity ("gravity") association by McNemar's test. The results of the association were compared pairwise. The proposed method corrects 666 associations of the baseline method, while failing in 280 cases where the baseline method is correct.

	Proposed: correct	*Proposed: failed*	*Row total*
Baseline: correct	2315	280	2595
Baseline: failed	666	1099	1765
Column total	2981	1379	4360

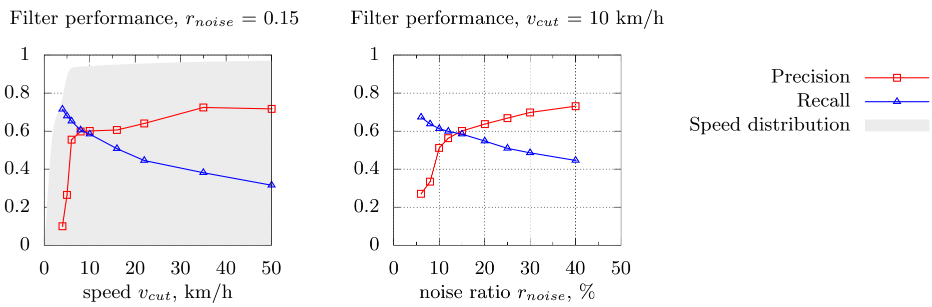

Figure 4: The performance of the noise filter on the synthetic dataset. Selecting the cut-off speed (left plot) and the noise ratio threshold (right plot) involves a compromise of precision and recall. In both cases there is a sharp drop off in precision when the filter becomes too aggressive. The grey background represents the cumulative distribution of speeds in the dataset.

row). In 280 cases the opposite effect was observed (column 3, top row). We evaluated the statistical significance with the mid-p-value method [8]. The estimate $p \ll 0.001$ suggests that the measured increase in accuracy is statistically significant.

5 CONCLUSIONS

We outlined methods that produce semantically annotated human mobility traces fro geo-tagged photos. We focused on two stages on this process: how the extraction of mobility traces can be improved by noise filtering and applying popularity data in semantic annotation.

Photos from sites like Flickr and Panoramio contain noise in the form of incorrect coordinates or time stamps. We have described a method that filters the noise by estimating the movement speed of the photographer. Whenever speeds above filter cut-off value are detected, the photo that is causing the effect is found heuristically and removed. While the filter attempts to detect time stamp noise, it performs much better on noise in coordinates. Until better methods are developed, we suggest external indicators to be considered for detecting time stamp noise.

We also showed that using a gravity-inspired method of place association, where more popular places exert their influence further, improves semantic annotation accuracy. We compared the method to the simpler association by proximity and measured a 9% improvement in accuracy on a real-world dataset. However, the absolute value of 68% accuracy of the automated annotation is still low. In future work, we aim to improve this area. Actual visited places are not pinpoint coordinates. We propose that instead of POIs, the photo traces, possibly integrated from multiple sources, should be mined for regions of interest (ROIs). This would allow semantic annotations with the extent and shape of the geographical features like buildings, bridges, parks or stadiums taken into account.

ACKNOWLEDGMENTS

This research has been supported by EU through European Regional Development Fund.

REFERENCES

[1] Natalia V. Andrienko, Gennady L. Andrienko, Georg Fuchs, and Piotr Jankowski. 2016. Scalable and privacy-respectful interactive discovery of place semantics from human mobility traces. *Information Visualization* 15, 2 (2016), 117–153. http://dx.doi.org/10.1177/1473871615581216

[2] Yuki Arase, Xing Xie, Takahiro Hara, and Shojiro Nishio. 2010. Mining people's trips from large scale geo-tagged photos, See [3], 133–142. http://doi.acm.org/10.1145/1873951.1873971

[3] Alberto Del Bimbo, Shih-Fu Chang, and Arnold W. M. Smeulders (Eds.). 2010. *Proceedings of the 18th International Conference on Multimedia 2010, Firenze, Italy, October 25-29, 2010*. ACM. http://dl.acm.org/citation.cfm?id=1873951

[4] Igo Ramalho Brilhante, José Antônio Fernandes de Macêdo, Franco Maria Nardini, Raffaele Perego, and Chiara Renso. 2015. On planning sightseeing tours with TripBuilder. *Inf. Process. Manage.* 51, 2 (2015), 1–15. http://dx.doi.org/10.1016/j.ipm.2014.10.003

[5] Satish Chandra and Anish Kumar Bharti. 2013. Speed Distribution Curves for Pedestrians During Walking and Crossing. *Procedia - Social and Behavioral Sciences* 104 (2013), 660 – 667. https://doi.org/10.1016/j.sbspro.2013.11.160

[6] Xihui Chen, Jun Pang, and Ran Xue. 2014. Constructing and Comparing User Mobility Profiles. *TWEB* 8, 4 (2014), 21:1–21:25. http://doi.acm.org/10.1145/2637483

[7] Munmun De Choudhury, Moran Feldman, Sihem Amer-Yahia, Nadav Golbandi, Ronny Lempel, and Cong Yu. 2010. Automatic construction of travel itineraries using social breadcrumbs. In *HT'10, Proceedings of the 21st ACM Conference on Hypertext and Hypermedia, Toronto, Ontario, Canada, June 13-16, 2010*, Mark H. Chignell and Elaine G. Toms (Eds.). ACM, 35–44. http://doi.org/10.1145/1810617.1810626

[8] Morten W. Fagerland, Stian Lydersen, and Petter Laake. 2013. The McNemar test for binary matched-pairs data: mid-p and asymptotic are better than exact conditional. *BMC Medical Research Methodology* 13, 1 (2013), 91. https://doi.org/10.1186/1471-2288-13-91

[9] Barbara Furletti, Paolo Cintia, Chiara Renso, and Laura Spinsanti. 2013. Inferring human activities from GPS tracks. In *Proceedings of the 2nd ACM SIGKDD International Workshop on Urban Computing, UrbComp@KDD 2013, Chicago, Illinois, USA, August 11, 2013*, Yu Zheng, Steven E. Koonin, and Ouri E. Wolfson (Eds.). ACM, 5:1–5:8. http://doi.acm.org/10.1145/2505821.2505830

[10] Fosca Giannotti, Mirco Nanni, Fabio Pinelli, and Dino Pedreschi. 2007. Trajectory pattern mining. In *Proceedings of the 13th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, San Jose, California, USA, August 12-15, 2007*, Pavel Berkhin, Rich Caruana, and Xindong Wu (Eds.). ACM, 330–339. http://doi.acm.org/10.1145/1281192.1281230

[11] Fabien Girardin, Filippo Dal Fiore, Carlo Ratti, and Josep Blat. 2008. Leveraging explicitly disclosed location information to understand tourist dynamics: a case study. *J. Location Based Services* 2, 1 (2008), 41–56. http://dx.doi.org/10.1080/17489720802261138

[12] Alexander Hinneburg and Hans-Henning Gabriel. 2007. DENCLUE 2.0: Fast Clustering Based on Kernel Density Estimation. In *Advances in Intelligent Data Analysis VII, 7th International Symposium on Intelligent Data Analysis, IDA 2007, Ljubljana, Slovenia, September 6-8, 2007, Proceedings (Lecture Notes in Computer Science)*, Michael R. Berthold, John Shawe-Taylor, and Nada Lavrac (Eds.), Vol. 4723. Springer, 70–80. https://doi.org/10.1007/978-3-540-74825-0_7

[13] Erik Jenelius and Haris N. Koutsopoulos. 2013. Travel time estimation for urban road networks using low frequency probe vehicle data. *Transportation Research Part B: Methodological* 53 (2013), 64 – 81. https://doi.org/10.1016/j.trb.2013.03.008

[14] Shan Jiang, Yingxiang Yang, Siddharth Gupta, Daniele Veneziano, Shounak Athavale, and Marta C. Gonzlez. 2016. The TimeGeo modeling framework for urban mobility without travel surveys. *Proceedings of the National Academy of Sciences* 113, 37 (2016), E5370–E5378. arXiv:http://www.pnas.org/content/113/37/E5370.full.pdf http://www.pnas.org/content/113/37/E5370.abstract

[15] Takeshi Kurashima, Tomoharu Iwata, Go Irie, and Ko Fujimura. 2010. Travel route recommendation using geotags in photo sharing sites. In *Proceedings of the 19th ACM Conference on Information and Knowledge Management, CIKM 2010, Toronto, Ontario, Canada, October 26-30, 2010*, Jimmy Huang, Nick Koudas, Gareth J. F. Jones, Xindong Wu, Kevyn Collins-Thompson, and Aijun An (Eds.). ACM, 579–588. http://doi.acm.org/10.1145/1871437.1871513

[16] Chunming Li, Yang Zhao, Xiaoze Sun, Xiaodan Su, Shuanning Zheng, Rencai Dong, and Longyu Shi. 2011. Photography-based analysis of tourists temporalspatial behaviour in the Old Town of Lijiang. *International Journal of Sustainable Development & World Ecology* 18, 6 (2011), 523–529. http://dx.doi.org/10.1080/13504509.2011.601471

[17] Quannan Li, Yu Zheng, Xing Xie, Yukun Chen, Wenyu Liu, and Wei-Ying Ma. 2008. Mining user similarity based on location history. In *16th ACM SIGSPATIAL International Symposium on Advances in Geographic Information Systems, ACM-GIS 2008, November 5-7, 2008, Irvine, California, USA, Proceedings*, Walid G. Aref, Mohamed F. Mokbel, and Markus Schneider (Eds.). ACM, 34. https://doi.org/10.1145/1463434.1463477

[18] Lin Liao, Dieter Fox, and Henry A. Kautz. 2007. Extracting Places and Activities from GPS Traces Using Hierarchical Conditional Random Fields. *I. J. Robotics Res.* 26, 1 (2007), 119–134. http://dx.doi.org/10.1177/0278364907073775

[19] Kwan Hui Lim, Jeffrey Chan, Christopher Leckie, and Shanika Karunasekera. 2015. Personalized Tour Recommendation Based on User Interests and Points of Interest Visit Durations. In *Proceedings of the Twenty-Fourth International Joint Conference on Artificial Intelligence, IJCAI 2015, Buenos Aires, Argentina, July 25-31, 2015*, Qiang Yang and Michael Wooldridge (Eds.). AAAI Press, 1778–1784. http://ijcai.org/papers15/Abstracts/IJCAI15-253.html

[20] Xin Lu, Changhu Wang, Jiang-Ming Yang, Yanwei Pang, and Lei Zhang. 2010. Photo2Trip: generating travel routes from geo-tagged photos for trip planning, See [3], 143–152. http://doi.acm.org/10.1145/1873951.1873972

[21] Jean Damascène Mazimpaka and Sabine Timpf. 2016. Trajectory data mining: A review of methods and applications. *Journal of Spatial Information Science* 2016, 13 (2016), 61–99.

[22] Anastasios Noulas, Salvatore Scellato, Neal Lathia, and Cecilia Mascolo. 2012. Mining User Mobility Features for Next Place Prediction in Location-Based Services. In *12th IEEE International Conference on Data Mining, ICDM 2012, Brussels, Belgium, December 10-13, 2012*, Mohammed Javeed Zaki, Arno Siebes, Jeffrey Xu Yu, Bart Goethals, Geoffrey I. Webb, and Xindong Wu (Eds.). IEEE Computer Society, 1038–1043. https://doi.org/10.1109/ICDM.2012.113

[23] Andrey Tietbohl Palma, Vania Bogorny, Bart Kuijpers, and Luis Otávio Alvares. 2008. A clustering-based approach for discovering interesting places in trajectories. In *Proceedings of the 2008 ACM Symposium on Applied Computing (SAC), Fortaleza, Ceara, Brazil, March 16-20, 2008*, Roger L. Wainwright and Hisham Haddad (Eds.). ACM, 863–868. http://doi.acm.org/10.1145/1363686.1363886

[24] Christine Parent, Stefano Spaccapietra, Chiara Renso, Gennady L. Andrienko, Natalia V. Andrienko, Vania Bogorny, Maria Luisa Damiani, Aris Gkoulalas-Divanis, José Antônio Fernandes de Macêdo, Nikos Pelekis, Yannis Theodoridis, and Zhixian Yan. 2013. Semantic trajectories modeling and analysis. *ACM Comput. Surv.* 45, 4 (2013), 42:1–42:32. http://doi.acm.org/10.1145/2501654.2501656

[25] Fabio Pianese, Xueli An, Fahim Kawsar, and Hiroki Ishizuka. 2013. Discovering and predicting user routines by differential analysis of social network traces. In *IEEE 14th International Symposium on "A World of Wireless, Mobile and Multimedia Networks", WoWMoM 2013, Madrid, Spain, June 4-7, 2013*. IEEE Computer Society, 1–9. https://doi.org/10.1109/WoWMoM.2013.6583383

[26] Georgios Skoumas, Klaus Arthur Schmid, Gregor Jossé, Matthias Schubert, Mario A. Nascimento, Andreas Züfle, Matthias Renz, and Dieter Pfoser. 2015. Knowledge-Enriched Route Computation. In *Advances in Spatial and Temporal Databases - 14th International Symposium, SSTD 2015, Hong Kong, China, August 26-28, 2015. Proceedings (Lecture Notes in Computer Science)*, Christophe Claramunt, Markus Schneider, Raymond Chi-Wing Wong, Li Xiong, Woong-Kee Loh, Cyrus Shahabi, and Ki-Joune Li (Eds.), Vol. 9239. Springer, 157–176. http://dx.doi.org/10.1007/978-3-319-22363-6_9

[27] Tanel Tammet, Ago Luberg, and Priit Järv. 2013. Sightsmap: Crowd-Sourced Popularity of the World Places. In *Information and Communication Technologies in Tourism 2013*, Lorenzo Cantoni and Zheng (Phil) Xiang (Eds.). Springer Berlin Heidelberg, 314–325. http://dx.doi.org/10.1007/978-3-642-36309-2_27

[28] Muhammad Reaz Uddin, Chinya V. Ravishankar, and Vassilis J. Tsotras. 2011. Finding Regions of Interest from Trajectory Data. In *12th IEEE International Conference on Mobile Data Management, MDM 2011, Luleå, Sweden, June 6-9, 2011, Volume 1*, Arkady B. Zaslavsky, Panos K. Chrysanthis, Dik Lun Lee, Dipanjan Chakraborty, Vana Kalogeraki, Mohamed F. Mokbel, and Chi-Yin Chow (Eds.). IEEE Computer Society, 39–48. http://dx.doi.org/10.1109/MDM.2011.12

[29] Hyoseok Yoon, Yu Zheng, Xing Xie, and Woontack Woo. 2012. Social itinerary recommendation from user-generated digital trails. *Personal and Ubiquitous Computing* 16, 5 (2012), 469–484. http://dx.doi.org/10.1007/s00779-011-0419-8

[30] Zengbin Zhang, Lin Zhou, Xiaohan Zhao, Gang Wang, Yu Su, Miriam J. Metzger, Haitao Zheng, and Ben Y. Zhao. 2013. On the validity of geosocial mobility traces. In *Twelfth ACM Workshop on Hot Topics in Networks, HotNets-XII, College Park, MD, USA, November 21-22, 2013*, Dave Levine, Sachin Katti, and Dave Oran (Eds.). ACM, 11:1–11:7. https://doi.org/10.1145/2535771.2535786

[31] Yu Zheng. 2015. Trajectory Data Mining: An Overview. *ACM TIST* 6, 3 (2015), 29:1–29:41. http://doi.acm.org/10.1145/2743025

[32] Yu Zheng, Licia Capra, Ouri Wolfson, and Hai Yang. 2014. Urban Computing: Concepts, Methodologies, and Applications. *ACM TIST* 5, 3 (2014), 38:1–38:55. http://doi.acm.org/10.1145/2629592

Appendix 3

Publication III

P. Järv, T. Tammet, and M. Tall. Hierarchical regions of interest. In *19th IEEE International Conference on Mobile Data Management, MDM 2018, Aalborg, Denmark, June 25-28, 2018*, pages 86–95. IEEE Computer Society, 2018

Hierarchical Regions of Interest

Priit Järv, Tanel Tammet, Marten Tall
School of Software Science
Tallinn University of Technology
Tallinn, Estonia
Email: priit@whitedb.org, tanel.tammet@ttu.ee, tall.marten@gmail.com

***Abstract*—Mining crowd-sourced movement trajectories is a useful tool in urban computing. Common mobility patterns of the visitors or residents of a city can be exploited in applications such as disaster management, transportation planning and ad placement. In recommendation systems, individual behaviour is of special interest. To extract the visiting behaviour of individuals, the trajectories need to be semantically annotated.**

We describe how hierarchical regions of interest (ROIs) can be used for semantic annotation. By combining multiple layers of smaller and larger regions we can flexibly detect both visits to dense hotspots and trajectory segments visiting larger areas, such as an old town, a park or an island. Extending the annotation beyond common hotspots captures more information about the behaviour.

I. INTRODUCTION

Inferring human activities from crowd sourced data can provide valuable data about behaviour. Studying mass movement of people, for example, is relevant when planning transportation networks, optimizing ad placement or preparing for disaster mitigation. The individual behaviour is useful in recommender systems. To recommend a place to visit or a sequence of activities, the system needs to be able to find those that the user would find interesting based on the user's earlier activities. We can make such predictions from the behaviour patterns that the previous visitors have exhibited.

Crowd sourced mobility data, such as GPS traces or geotagged social media contributions, are usually in the form of raw trajectories. Semantic annotation transforms sequences of points into sequences of activities. This can be done by assigning place names like "Eiffel Tower" or descriptive tags like "tower, view, ..." to points. In this paper we will focus on describing trajectories in terms of what places they visit. Understanding the visitors activities then depends on the ability to connect an arbitrary location to a known place name.

We loosely define the terms "place", "area" and "region" as they will be used in the paper. When we refer to "places" we mean geographical objects in general that are used, visited and recognized by people. Similarly, "place name" is something that universally identifies the place. We use "places" and "areas" in the same context when we want to make the distinction in between smaller and larger geographical objects. Finally, the term "region" is used exclusively to mean a map polygon.

The most common spatial representation for the interesting places is a point, such that the term "point of interest" sometimes can be synonymous with a coordinate pair attached to a place name. In geographic information systems, various geometric shapes, including lines and polygons are used. Data mining human mobility data can reveal regions, also representable by polygons, which do not necessarily coincide with geographic information and are called regions or areas of interest. Throughout the paper, when we use the term points of interest (POIs), we mean the representation of interesting places as named x, y coordinate pairs. When we use the term regions of interest (ROIs), we mean the representation of the same places or areas as named polygons.

The basic method of associating trajectories with visited places is to find a suitable nearby POI for each segment or stay point. When there are multiple POIs nearby, selecting the correct one is complicated, because the POI representation does not encode any information about the shapes and sizes of places. In contrast, with ROIs we can find the place name of an arbitrary location by checking whether the coordinates intersect any ROI (Fig. 2). In case of multiple intersections, the decision is still straightforward if the ROIs have a clearly defined relationship (such as one being contained in another). Such relationship is described by a hierarchy of ROIs.

Places form a natural hierarchy. A church may belong to a historic area that in turn is a part of some city district. A sculpture may be located in a park with multiple attractions and the park itself may be part of a thematic neighbourhood. Studies about ROIs are generally either about popular hotspots or larger thematic areas. In recommender system papers, the focus has been on hotspots, but this would mean that part of the information that could be available is consciously removed in the semantic annotation process. In other words, the areas where stay points are sparse are treated as "nothing to see here". In case of geotagged photos, this contradicts the usual purpose of sharing photos on social media - the person who uploaded the photo has considered the place to be visually interesting to other people.

We propose to extend the notion of a flat representation of the city space into a natural one, where multiple layers of ROIs are arranged hierarchically. This retains the ability to recognize hotspots vital to recommender systems, but

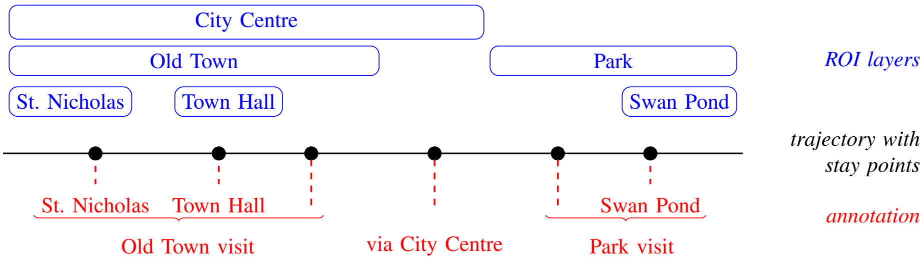

Figure 1. Trajectory annotation using regions of interest (ROI) layers. The higher layers allow annotation of the middle trajectory segment and provide context to individual place visits.

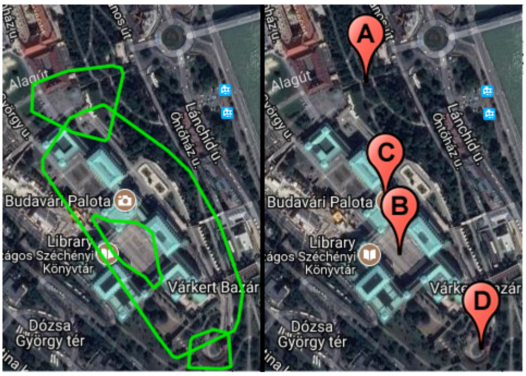

Figure 2. Expressive power of representing places as regions of interest (ROI; shown left) compared to points of interest (POI; right). Pictured here is a scenic lookout at the top of Budapest Funicular (A), Buda Castle courtyard (B), Buda Castle (C) and Rondella (D). The regions designated by polygons allow determining the place name for an arbitrary point by finding which region it intersects. Map data by Google, CNES/Airbus, DigitalGlobe.

also allows recognizing generalizations, thematic areas or simply large features like parks or nature reserves that could remain undetectable to data mining techniques due to the spatial sparsity of visitors' activities. This flexibility is the main motivator for designing the proposed approach and also results in a directly measurable effect of being able to meaningfully annotate more stay points on trajectories.

ROI naming is as important as ROI discovery. While the hierarchy of ROIs itself already contains valid information about the behaviour, it is difficult to produce generally useful output unless the ROIs represent real-world places. For example, when using ROIs to recommend a place to visit, the output should be something that can be enriched with external descriptions, pictures and that the user recognizes or can look up easily. Naming ROIs is not trivial, as we will illustrate by our experiments. Therefore, we propose that the naming process will let a human expert review the most important decisions. The majority of regions will still be named fully automatically, but those that affect the accuracy of the annotation most and/or are most difficult to decide will be presented to the human expert.

In the paper, we describe the approach from a general perspective and also give a technical overview of a proof of concept application - annotating individual movement trajectories of sight-seeing trips, extracted from geotagged photos (Fig. 1).

The remainder of the paper is organized as follows. In Section II we review the related work. In Section III we describe how hierarchical ROIs can be formed and used to semantically annotate trajectories. In Section IV we show how the accuracy of the semantic annotation is improved using our implementation of the hierarchical ROIs. We use Flickr photos to find the ROIs and then annotate sightseeing trajectories extracted from a different source – Panoramio photos – with suitable place or area names. We also give the results of a survey conducted to evaluate the quality of the region shapes. Section V summarizes the findings and suggests directions for future research.

II. Related Work

One of the earliest works to describe both the discovery of ROIs and their use in extracting behavioural data from trajectories was by Giannotti et al. They represent ROIs as rectangular shapes [1]. The method of Giannotti et al. has been enhanced later, for example by improving ROI boundary detection by pre-filtering stay points with low local density [2]. Extracting interesting regions by applying density-based clustering to individual stay points in GPS traces has been proposed in several papers [3], [4], [5]. The regions are represented as a set of stay points which effectively correspond to an arbitrary shape. Uddin et al. describe an efficient method to find regions as the set of points that have many trajectories of slowly moving objects passing nearby [6].

Discovery of ROIs from geotagged photos has been mainly explored in the context of sightseeing recommendations. Kisilevich et al. adapted DBSCAN specifically for this purpose, by defining density as the minimum number of distinct users in the neighbourhood and introducing adaptive density threshold for splitting high density clusters by local variations. The modified algorithm is called P-DBSCAN [7]. Liu et al. modify density based clustering by introducing a predefined order in which photos are processed, so that points are assigned preferentially to clusters where there are also more popular Foursquare venues nearby [8]. Laptev et al. proposed a grid-based method, related to the Gaussian smoothing and watershed segmentation image processing techniques [9]. Their method takes the desired region size as a parameter and automatically adjusts the kernel bandwidth used in Gaussian density estimation to produce the clustering. Cai et al. use a grid-based method where the density of cells is defined as the number of intersecting trajectories [10]. Shirai et al. include the angle of view and orientation in extended photo metadata to infer the shapes of places [11].

The concept of hierarchical ROIs was introduced in the GeoLife application for finding similar users and making location recommendations [3], [4], [12]. The method is based on GPS traces and shares many similarities to the approach we describe in this paper. The main difference is that their ROIs represent latent features of the visitor behaviour, while we attempt to ultimately explain trajectories as sequences of known place names.

Semantic annotation involves enriching the trajectory segments with various labels, such as "stop" and "move" episodes, mode of transportation, place visit or activity [13]. In the following we review papers related to place association, particularly in the context of recommender systems.

The simplest technique of annotating photo based traces is to associate each stay point with the nearest POI within a given radius [14], [15], [16], [17]. This method requires however, that only POIs relevant to the application are included. A variation of this method is to include POI popularity when estimating the likelihood that it was visited [18].

Annotating with ROIs can be accomplished by finding spatial overlap of the stay point and a ROI [19]. In case of GPS traces, the visit of a place can also be defined as the intersection of a trajectory and a ROI with a long enough duration [20].

When using automatically discovered ROIs, the problem of finding the semantics of the ROIs has to be solved. Crandall et al. showed that for a place represented by a set of photos, simply selecting the most distinctive tags can generate accurate labels [21]. Yin et al. apply a generative mixture model using the sets of Flickr photo tags in a location and the overall distribution of tags [22].

In this paper, we also discuss extracting a hierarchy of significant clusters from a dendrogram representation. To the best of our knowledge, general purpose methods that select significant clusters while maintaining the hierarchy relation have not been studied extensively. Sander et al. present a method that recursively splits the points to clusters at significant local maxima of a reachability plot. Dendrograms are handled by converting them to reachability plots [23]. Campello et al. describe both a simplification of the dendrogram by setting a minimum cluster size and an optimal method of creating a flat clustering based on cluster stability [24].

III. METHODS

Conceptually, describing places as a hierarchy of regions of interest (ROIs) and semantic annotation of trajectories can be based on different kinds of spatial data, such as GPS traces. The requirement is that we can establish both frequently visited places and stay points on individual trajectories.

In this paper, we use geotagged photos for both finding ROIs and individual trajectories. The crowd-sourced photos from Flickr (http://flickr.com) and Panoramio (closed in 2016) are attractive for this purpose because of universal availability and rich semantic meta data – tags and titles.

Semantic annotation with ROIs consists of three stages: finding the boundaries of ROIs, associating each ROI with a place or an area and annotating trajectory segments based on which ROIs they intersect. We will refer to the proof of concept method presented in this paper that includes all three stages as the "Hierarchical ROI", or HROI method. While we present a specific implementation, there is flexibility in choosing the exact technique used in each stage.

A. Region shape formation

We start with the assumption that interesting places are those where the density of photos is higher. The assumption relies on the data being sourced from an application where the focus is on presenting interesting imagery to the general public (Flickr, Panoramio).

We find the ROIs by clustering the photos. The simplest approach to creating a hierarchy of ROIs, or layers of larger and smaller ROIs would be to apply a clustering algorithm once for each layer, adjusting the parameters so that clusters of appropriate sizes and densities are created. The main challenge with this approach is determining the parameters and coping with local density variations. The difference in density between the tourist attractions in a city can be big.

To address the issue of local density we chose HDBSCAN as the clustering method. HDBSCAN is a hierarchical method and produces a dendrogram of clusters. In the dendrogram, clusters of varied densities can co-exist in separate branches. To extract a hierarchy with a fixed number of layers we use a similar approach as described in [24]. We will refer to the method of creating n levels of clusters as HDBSCAN/n.

Let $G = (V, E)$, where V is the set of nodes and E is the set of edges, represent the simplified dendrogram as defined by Campello et al. We set the node weight $w(v)$ for all

$v \in V$ to be the stability of the cluster represented by the node. We define the n-layered clustering as the tree $G_n = (V_n \subset V, E')$ such that the maximum depth of $G_n < n$ and $\sum_{v \in V_n} w(v)$ is maximised. In other words, we make a tree with n levels by selecting the nodes in G so that we get the maximum total cluster stability. The set V_n can be constructed by recursive depth first search.

function MAX_WEIGHT_SUBSET(n, v_{root})
  Initialize $S_j \leftarrow \emptyset$, $j \in 1, \ldots, n$
  for each child v_i of v_{root} **do**
    $S_1^i \ldots S_n^i \leftarrow$ MAX_WEIGHT_SUBSET(n, v_i)
    for $j \leftarrow 1, n$ **do**
      $S_j \leftarrow S_j \cup S_j^i$
    end for
  end for
  Find smallest k such that $w(v_{root}) > \sum_{v \in S_k} w(v)$
  $S_{j+1} = S_j$ for $j \geq k$
  $S_k = \{v_{root}\}$
  Return $S_1 \ldots S_n$
end function
$S_1 \ldots S_n \leftarrow$ MAX_WEIGHT_SUBSET(n, root of G)
$V_n \leftarrow \bigcup^n S_j$

The depth first traversal builds V_n recursively by using sets $S_1 \ldots S_n$ to simultaneously keep track of nodes that would be included in $V_1 \ldots V_n$. The algorithm relies on the idea that for any n less than the current maximum depth, when a parent node is encountered, either the parent is included in V_n and some nodes from each subtree are removed from V_n, or the parent is excluded and the results from the subtrees are joined to make the new V_n. The decision is made depending on which results in the new V_n with a larger total weight.

This algorithm runs in linear time $O(|V|)$ as each node in G needs to be visited once and n is constant. To find E', we do a second depth first traversal and add an edge from each node in V_n to the next node on the path from root in G that is also in V_n. This will reconnect the hierarchy in places where lower stability nodes were removed.

Proof sketch. We show that the constructed set V_n has the maximal total weight for a given G and n.

1) Assume that there is a tree G with maximum depth d. This tree is trivially the same as G_{d+1}. Divide the set of nodes V into disjoint subsets S_j, $j = (1 \ldots d+1)$ such that (a) none of the sets contain two nodes that lie on the same path from the root node; and (b) each S_j forms a maximal set such that we can construct the maximal $V_n = \bigcup^n S_j$.
2) Show that the properties from Step 1 are maintained when we connect m parallel subtrees of maximum depth $d-1$ satisfying the conditions in Step 1, to a root node v_{root} to form a new tree of maximum depth d. We find new sets $S_k = \bigcup_i^m S_j^i$ where if $\sum_{v \in S_k} w(v) > w(v_{root})$ then $k = j$, otherwise $k = j+1$. Property (a) is maintained because the sets S_j^i come from disjoint subtrees and the new shared root will be assigned to its own set $\{v_{root}\}$. Assume v_{root} is part of V_n. In this case, property (b) is maintained because if we construct V_{n-1}^i separately for each subtree it will be maximal. If v_{root} does not belong in V_n then property (b) is again maintained because each subset V_n^i in $V_n = \bigcup_i^m V_n^i$ is also maximal so $\sum_{v \in V_n} w(v)$ is maximal.
3) For $d = 1$, Step 1 and Step 2 can be demonstrated trivially.

To assist in finding larger semantically similar regions, we use a distance function which combines spatial distance, similarity between the Flickr tags of photos and the geographical area (Flickr's "Where on Earth" (WoE) id). Additionally, to facilitate cluster boundary formation at physical obstacles, such as cliffs, city walls, bodies of water or motorways, we use the Open Source Routing Machine (OSRM) [25] to calculate spatial distance.

The distance between two photos x_i and x_j is a linear combination of spatial distance with routing d_{osrm}, Jaccard distance between the sets of tags T_i, T_j of the two photos $d_J = 1 - \frac{|T_i \cap T_j|}{|T_i \cup T_j|}$ and the distance between the WoE ids of the two photos d_{WoE},

$$d_s(x_i, x_j) = d_{osrm}(x_i, x_j) + \beta d_{WoE}(x_i, x_j) + \gamma d_J(x_i, x_j)$$

In this paper, we simplify d_{WoE} so that it is 1 when the two photos share the same lowest level WoE id and 0 otherwise. The parameters β and γ determine the influence of the geographic region and tag similarity, respectively.

Because of the distance function, the clustering happens in a non-metric space, meaning that spatial indexes are not available for efficient neighbourhood queries. In practice, to avoid repeated computation of pairwise distances, we run the Python implementation of HDBSCAN algorithm [26] on a k-neighbourhood graph. The complete pairwise distance matrix would require $O(n^2)$ storage, which is impractical for even modest size input datasets (100000-650000 photos). The neighbourhood graph reduces the storage requirement to $O(kn)$. To reduce the computation time of the neighbourhood graph, we only consider spatially close points as neighbour candidates. We use a ball tree index with haversine metric. The haversine distance d_h is effectively a lower bound on our distance function d_s, meaning that for each point x_i, assuming that some point x_j is currently the k-th closest neighbour, we only need to compute pairwise distances to points where $d_h(x_i, x_{candidate}) < d_s(x_i, x_j)$.

We create the boundaries of ROIs by drawing an alpha shape (concave hull) [27] around the set of points, or photos, in each cluster. Since it is reasonable to assume that the object extends somewhat beyond the outermost photos, we move each boundary point in the direction opposite to the cluster centroid, such that it is 10% further away from the centroid.

B. Region names

Before the ROIs can be used in semantic annotation, we must discover what real-world places they represent. We use both the meta data of the photos that formed a region and geographical information – a POI database. The tags and titles of photos may contain information about the place. Additionally, we must consider all POIs located within the region boundary and nearby as potential sources for the place name.

We base the method of name finding on information retrieval techniques. We treat the regions as documents, semantic meta data as terms and possible name candidates as queries. We can then use latent semantic indexing (LSI) [28], which calculates term-based similarity between queries and documents, to find the similarity between name candidates and regions. We create a shortlist of semantically related name candidates for each region and then do final name resolution using several ad hoc heuristics. The name resolution process attempts to detect non-ambiguous cases where, for example, a local POI is better than other name candidates.

Larger and denser regions are difficult to name automatically because of abundance and diversity of places and themes within. At the same time, from the perspective of semantic annotation, accurate naming of these regions is most important, since the trajectories will intersect these regions most often. Because a small number of region names can be validated with little effort, we complete the name resolution with a human-assisted process, where our program displays the region boundary, name candidates with semantic similarity and other meta-data and allows a human expert to either accept or change the name. This is done for a subset of regions, the remainder will keep their automatically assigned names.

The technical description of region naming process is as follows:

1) Download POIs located in areas covered by the regions. We create a grid of points and use Foursquare venue search for each point. We use coarse category-based filtering to only include places relevant to the application (in the context of our experiment, annotating sightseeing trips).
2) Create a list of terms appearing in photo tags, titles and POI names. We filter irrelevant terms, such as common words, image file names and terms used by only a single user. We create a translation dictionary for words that are similar by Jaro-Winkler distance [29] to further reduce the set of terms.
3) Create a term-document matrix by representing each ROI as a vector where each element is the number of distinct users who tagged their photos with the term. We then apply the TF-IDF and LSI transformations from the Gensim package [30] to create a term-document representation with reduced dimensions.

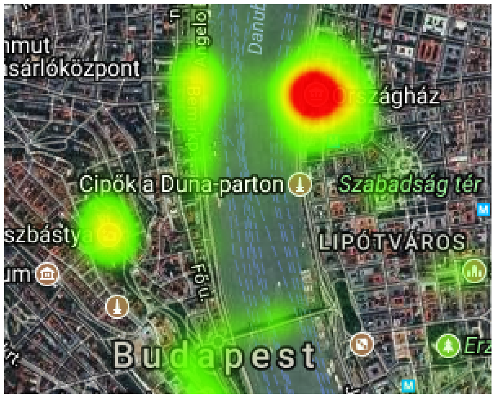

Figure 3. Heat map of the distribution of photos tagged with *parliament*, or a synonymous term. Several locations across the Danube river provide a view of the building and the corresponding regions have a strong semantic similarity to the actual parliament area. In this case, such regions can be correctly described as "view points", but in general semantic similarity to a remote object may also occur if there are no distinctive local features in the region. Map data by Google.

4) Create name candidates from photo titles and from all POIs in the area of regions. In each region, we find all n-grams of words in photo titles where $1 \leq n \leq 5$ and select one with the most distinct users for each value of n. We select a few additional n-grams where $n > 1$ that are sufficiently different from the already selected ones (edit distance of word sequences of at least 2). All POI names are also added to the set of name candidates.
5) We calculate the similarity of between each name candidate and ROI. Each name candidate is normalized using the translation dictionary. The words that do not appear in the dictionary are removed. Candidates from POI names have the venue's main category appended. We then convert each candidate to a vector in the term-document space and find cosine similarity $s_{candidate}$ to each ROI using the Gensim package.
6) Apply several ad hoc heuristics to detect non-ambiguous cases, where a single candidate is much better than other names. In case there is no single preferred name candidate, the one with the highest weighted semantic similarity $(1-\alpha)s_{candidate} + \alpha s_{candidate}\frac{\ln N_{candidate}}{\ln N}$ to the region is chosen. In case of title-derived names, $N_{candidate}$ is the number of distinct users of the n-gram, while N is the maximum number distinct users for any n-gram over the entire set of ROIs. In case of Foursquare POI names, these values instead represent the number of users checking in the venue. The name resolution procedure also attempts to flag cases where there is

a potential problem with the name candidate. Flagged cases include when the selected name is a 1-gram, when there was no single preferred name candidate and also when the name was taken from a POI not within the region boundary (Fig. 3).

7) Perform human-assisted name resolution for the following cases: the ROIs with most photos, ROIs that are both large and dense (sorted by rM where r is the distance between region centroid and its furthest point and M is its number of photos) and finally remaining ROIs with high number of photos that have been flagged by the automatic name selection process.

We find that the human-assisted name resolution is not labour intensive, and has a high impact on the accuracy of the semantic annotation. We chose 30 each of popular, large and flagged ROIs (as described in step 11). For an expert familiar with the city, the name resolution took less than 2 hours. However, this approach does not scale beyond city scale.

C. Semantic annotation

The named regions created in previous two stages allow mapping any point within their boundaries to a place name trivially. The annotation of trajectories involves finding the stay points on trajectories and selecting the best matching region for each. In case of trajectories derived from geotagged photos, we consider clusters of spatio-temporally close photos as stay points. For GPS trajectories, various methods have been discussed in a trajectory data mining review by Zheng [31].

If the stay point lies within multiple regions, we choose the region from the lowest possible layer in the cluster hierarchy. In other words, we always attempt to find a place-level region, but if that is not available we fall back to an area-level region.

Since the region boundary is formed around the outer points in the region and then expanded further outward, it is possible that regions on the same layer overlap. In this case we have two tie-break criteria – the confidence that the ROI has a correct name (represented by the weighted semantic similarity as described in III-B) and the distance of the annotated point to the centroid of each region.

The overlapping can also occur, when we have multiple "themes" in a region, which increases separation in the space defined by the distance function d_s. This may cause geographically adjacent points to be assigned to different clusters. While this corresponds to the reality that same regions may have different uses and attractions for different visitors, the simple annotation approach described here does not exploit this information (Fig. 4).

IV. Results

We evaluated the viability of the hierarchical ROIs by testing the following:

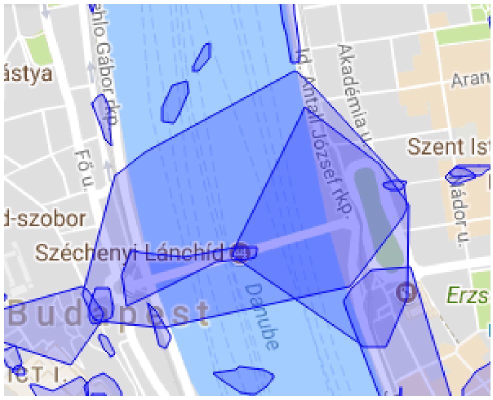

Figure 4. Spatially overlapping regions belonging to a single layer in regions of interest (ROI) hierarchy. This is partially caused by clustering with a distance function that includes semantic distance. Two points at the same geographical location may appear as apart from each other to the clustering algorithm. The effect is undesirable, unless semantic similarity to the region is also considered in trajectory annotation. Map data by Google.

1) How well does the method perform in its intended application, semantic annotation of trajectories? We measured quantitatively the accuracy of place name association to sightseeing trajectories extracted from Panoramio photos.
2) How accurate are the formed regions and how well do they cover interesting places? We conducted a survey and asked the participants to qualitatively judge the region shape and placement.

A. Annotation accuracy

The annotation experiment was performed in four European cities: Budapest, Tallinn, Venice and Vienna. We used Flickr photos taken in the area of these cities to find the ROIs, Google Places and Foursquare venues services as POI databases.

We downloaded photograph metadata from Flickr, generated hierarchical clusters and assigned region names as described in Section III. We adjusted the distance function for each city by visually inspecting the resulting clustering and checking that both individual places and important larger areas are represented. While this approach does not guarantee optimal performance, numerically optimizing the parameters is not viable, because the accuracy of the final results depends also on region naming, which in turn depends on cluster sizes and shapes. The chosen distance function parameters are given in Table I. The same process was also used to qualitatively determine the suitable number of layers. We chose 3 layers, as this was the minimum number to obtain both fine detail and larger area formation in the clustering. After creating the ROI boundaries, we found

Table I
DISTANCE FUNCTION AND CLUSTERING PARAMETERS

	HROI			*P-DBSCAN*		
	β	γ	*cluster size*	ϵ	*MinOwners*	*Addt*
Budapest	30	10	25	50	5	0.8
Tallinn	20	10	10	70	10	0.8
Venice	8	5	60	30	15	0.9
Vienna	20	10	60	30	15	0.9

the place and area names they represent as described in Section III-B.

We compared the method to four other techniques. For a naive baseline method, we implemented proximity based association, where the closest POI is always selected. For each annotated stay point, we made a Google Places query in the 200m radius of the point. The Google Places service is general purpose and therefore returns many POIs irrelevant in the context of sightseeing. We added type-based filtering, but this is not enough to work around this issue because the type metadata provided by the API is not accurate enough.

Google Places service also offers ranking places by prominence. For the second baseline method, we used this parameter in the nearby places query and picked the top-ranked POI after type filtering. This method preferentially picks a *point_of_interest* type POI, unless none are present in the first page of results.

The Google Places POI association methods represent minimum effort approaches using a public database. We also implemented a method that, to the best of our knowledge, represents the state of the art in annotating trajectories using a general purpose database of POIs. We queried the Foursquare API for venues in 200m radius of the stay point. We filtered the venues by high-level categories to remove non-sightseeing places and chose a venue l that had the highest "gravity" – measure of local influence proportional to the popularity of the venue and decreasing proportionally to the square of the distance to the venue r_l. We calculated the "gravity" by $\frac{users_l^g}{r_l^2}$ where $users_l$ is the number of users that have checked in to the venue. We used $g = 0.5$ as a normalization parameter, guided by the relative radius of influence the most popular venues were expected to have [18].

Finally, we implemented a non-hierarchical ROI based method to measure how much the proposed use of multiple layers contribute to the results. We chose P-DBSCAN as it is specifically designed to extract ROIs from geotagged photos. We chose *MinOwners* and ϵ parameters such that ROIs also appear in lower density areas, and a high adaptive density threshold *Addt* to achieve segmentation of dense groups of places. In some cities, we had to lower the neighbourhood size parameter ϵ to break apart dense central areas, because raising *Addt* very high would make the algorithm very sensitive to variations in local density. Table I gives the chosen parameters. We used the method described

Table II
ANNOTATION ACCURACY

	Budapest	Tallinn	Venice	Vienna
Sample size	517	553	511	522
	place level, %			
GP proximity	31	29	24	34
GP prominence	27	7	6	38
4sq gravity	56	51	36	51
P-DBSCAN	37	6	27	57
HROI	50	46	35	52
	place or area level, %			
GP proximity	41	32	31	49
GP prominence	35	9	10	54
4sq gravity	68	67	60	68
P-DBSCAN	83	94	76	78
HROI	74	82	69	78

in Section III-B to add place or area names to each ROI created by the P-DBSCAN algorithm.

We measured the annotation accuracy by generating sightseeing trajectories in each of the cities by filtering users who only took photos during a short period (no more than 21 consecutive days). The photos were split into daily trips based on their timestamps. Each trip was then filtered to remove photos with detectably invalid coordinates or timestamps and transformed into a sequence of stay points [18].

For Budapest, Venice and Vienna we used Panoramio photos as the source for trajectories to annotate. This avoids possible over fitting bias with the regions and trajectories originating from the same photos. For Tallinn we used Flickr photos as a suitable Panoramio dataset was not available, due to the closure of the service. In each city we annotated all stay points using all five compared methods. We then manually checked the correctness of annotations. Manual validation requires examining the relevant photos, local map, online travel guides, aerial imagery and other information. We validated a randomly chosen sample of stay points in each city.

For each stay point we rated each algorithm as "inaccurate", "area level accurate" or "place level accurate". We considered the annotation accurate, if the named place was the subject of the photo(s), the name was used in the photo title(s) or at least one photo was taken very close to the named place. In case of photo subjects, we also assessed whether the subject was close and accessible to the photographer on foot. If the annotation allowed identifying the surrounding area rather than the exact location, we rated the annotation "area level accurate". In case of missing annotations, the rating was automatically "inaccurate". Table II gives detailed results for both categories of accuracy. *Sample size* is the number of randomly selected annotations that were manually validated. We abbreviate the sources of POI data as follows: Google Places as "GP" and Foursquare as "4sq".

Examining the results by city, some features of the algorithms are revealed. The Google Places based methods

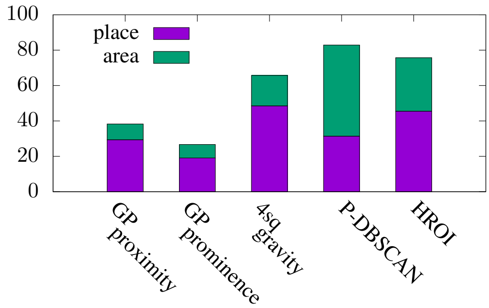

Figure 5. Overall annotation accuracy. The bars represent the percentage of places annotated accurately at place level and at area level. The HROI method provides the best compromise between covering a higher ratio of stay points and maintaining place level accuracy.

depend on the spatial distribution and type of POIs in the city, the "prominence" method additionally on the unknown algorithm for ranking prominence. Differences by country or city cause a lot of variance in the result.

P-DBSCAN generally performs well, but the results for Tallinn show that it can sometimes be difficult to find a good clustering with just one layer. Here the reason may be that most of the sightseeing activity in Tallinn is concentrated in just one area, the Old Town, which is much denser than the other sightseeing places. Hence the clusters in Old Town are either too large or the clusters outside of Old Town mostly disappear.

In Venice photogenic places fill a high density area of roughly $10km^2$, which is challenging both in terms of clustering and finding terms distinctive to a region. We also found that several Foursquare POIs had incorrect coordinates, which was not observed in other cities. This impacts the gravity-based method directly, but ROI based methods also relied on Foursquare. Finally, unlike other cities, many photos were taken from waterways. From a sightseeing point of view, water-based visits and land-based visits cannot be mixed arbitrarily in annotations. Because our method does not distinguish between visiting modes, we had to count stay points on water as incorrect, unless the annotated name stated both the correct location and that it was on water.

We summarize the results over all cities in Fig. 5. The gravity based POI association method achieved the highest annotation accuracy where the correct place name was chosen. The P-DBSCAN single layer ROI method was able to add a correct annotation the most points. However, the HROI method does exhibit the flexibility that was the main goal of developing the hierarchical approach. It performed well in terms of both finding accurate place names and annotating more stay points.

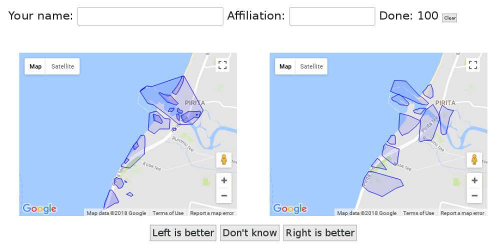

Figure 6. Application to compare the shapes and placement of regions. The participants were presented random map views with a random pairing of algorithms and asked to choose which of the two maps had better region boundaries. Map data by Google.

B. Region shapes

To evaluate different methods of discovering ROI boundaries, we asked survey participants to qualitatively assess coverage and accuracy of regions when displayed on the map. The participants were presented with regions created by two algorithms side by side (Fig. 6). The map view and the algorithms were selected randomly. The participants then had to select, which map had more places marked accurately. To do so, they were instructed to count the places by using the following accuracy scale:

- (count as 1) a region roughly matches one place.
- (count as 1) a region matches one place and has smaller subregions inside
- (count as 0.5) a region covers a place but extends far away from it
- (count as 0.5) a place is partially covered by one or more small, fragmented regions
- (count as 0) even after investigation, you are unsure what the region covers

Each time there was also the option to select "Don't know".

In this experiment, we compared the HDBSCAN/3 clustering method with P-DBSCAN and the region discovery method of Laptev et al. [9]. With both of these algorithms, we generated a set of ROIs with a single layer and a set of ROIs with three layers. The regions were generated from geotagged Flickr photos in Tallinn.

We used ϵ values 110, 50 and 30; $MinOwners$ values 7, 7 and 30; and $Addt$ values 0.5, 0.5 and 0.85, respectively, to create three layers with the P-DBSCAN algorithm. The parameters were chosen empirically to achieve coverage of the less dense map periphery and segmentation of the dense centre. We chose the settings of the middle layer for the single layer version. We used haversine distance in clustering.

The Laptev et al. algorithm has an intuitive *area* parameter that controls the size of generated regions. We used *area* of $0.01km^2$, $0.1km^2$ and $1km^2$ for the three-layered version and $0.1km^2$ for the single layer, as in the original paper. The grid size $K = 1024$. Because this algorithm is grid based, defining a distance function is not required.

Over 300 residents of Tallinn participated in the survey, with 17352 comparisons being included in the final results. The results did not establish a clear dominance of any single method. The following list gives the ratio of ”wins” in a pairwise comparison for each algorithm:

1) P-DBSCAN, 3 layers: 0.48
2) HDBSCAN/3: 0.48
3) Laptev et al., 3 layers: 0.46
4) Laptev et al., 1 layer: 0.41
5) P-DBSCAN, 1 layer: 0.35

The participants rated hierarchical methods higher than single-layered ones. In cases where a 3-layered method was paired against a 1 layer method, the hierarchical method was chosen in 51% of the cases. In 13% of the cases, the participant selected ”Don't know” and in 36% of cases they preferred the flat method.

Both state of the art algorithms performed well in this experiment so we also discuss the practical aspects of their potential use in generating hierarchical ROIs. The Laptev et al. algorithm is easy to implement with Python SciPy and scikit-image libraries. It was fast, completing the computation in under 1 minute for all datasets (here and in the following, we report the runtime of a Python implementation running on a single core of a Xeon E5-2690). There is one parameter that directly controls final cluster size. However, it has not been established yet how well this algorithm performs with different values of the *area* parameter.

P-DBSCAN is also straightforward to implement and the runtime was approximately 2 minutes on the largest dataset. The adaptive density threshold is well suited for breaking apart dense areas with many places. Its weakness is that it will require three parameters that need to be chosen for each layer, although there are some guiding principles. There are also limits to the variation in local density it can cope with.

The main advantage of HDBSCAN/3 compared to the other two algorithms is that the creation of multiple layers is data driven. While it did not outperform other methods in this experiment, in Section IV-A the performance was among the most stable across different datasets. The chosen distance function however made practical use difficult. Over 95% of computation time was spent in calculating the sparse neighbourhood graph, with total run times ranging from 5 to 10 hours. The distance function has parameters that are not easily interpreted.

V. Conclusion

Associating trajectory segments with visited places using POIs (as represented by points) is a common approach in research papers, but the problem of selecting the correct POI is not trivial. The popularity of the places, as indicated for example by Foursquare check-ins, is a heuristic that performs well, but still does not take into account the shape and size of the physical places.

We proposed that the annotation is done using regions of interest, or ROIs, that are capable of representing these aspects of places. The natural extension of this approach is that the ROIs are arranged as a hierarchy, so that both large areas and individual places can be represented simultaneously. This allows explaining additional parts of trajectories that do not match any hotspots and finding cases where individual visits belong to a single theme (such as a historic area).

We described an implementation of the hierarchical ROI, or HROI method. It consists of finding region boundaries, selecting names for the regions and finally matching appropriate regions to trajectory segments. While the region naming is an additional obstacle that is not encountered with pre-existing POI databases, human assisted name resolution of the most important ROIs is a low effort, high positive impact improvement in this task. Our method transforms sequences of stay points into sequences of known places.

We used the HROI method to annotate sightseeing trajectories with place names. We showed experimentally that place level accuracy with HROI was similar to the best POI based method, but thanks to the additional area level annotations HROI was consistently able to explain more stay points correctly. Overall we were able to annotate 76% of the stay points on sightseeing trajectories in Budapest, Tallinn, Venice and Vienna.

We also arranged a survey to qualitatively judge the size and placement of regions generated in the first stage of the HROI process. We compared the data driven hierarchical method HDBSCAN/n with state of the art methods of discovering ROIs from geotagged photos. The survey participants preferred hierarchical regions to flat ROIs in terms of how many interesting places they covered accurately.

The semantic- and street network aware distance function used with HDBSCAN/n was found to be slow and complex to use due to introducing additional parameters. Future work should examine whether a conventional spatial distance function with metric properties would achieve similar results.

While GIS data is less suitable for semantic annotation on detailed level (i.e. the name of a museum is more descriptive than its street address), using reverse geocoding to assist in discovering the names of large scale features should be investigated.

Acknowledgement

The authors would like to thank the HITSA foundation for funding the hardware used in this paper through the ”IT Akadeemia” programme. We also thank the reviewers for insightful comments and corrections.

References

[1] F. Giannotti, M. Nanni, F. Pinelli, and D. Pedreschi, "Trajectory pattern mining," in *Proceedings of the 13th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 2007, pp. 330–339.

[2] X. Chen, J. Pang, and R. Xue, "Constructing and comparing user mobility profiles for location-based services," in *Proceedings of the 28th Annual ACM Symposium on Applied Computing, SAC '13*, 2013, pp. 261–266.

[3] Q. Li, Y. Zheng, X. Xie, Y. Chen, W. Liu, and W. Ma, "Mining user similarity based on location history," in *16th ACM SIGSPATIAL International Symposium on Advances in Geographic Information Systems, Proceedings*, 2008, p. 34.

[4] Y. Zheng, L. Zhang, X. Xie, and W. Ma, "Mining interesting locations and travel sequences from GPS trajectories," in *Proceedings of the 18th International Conference on World Wide Web, WWW 2009*, 2009, pp. 791–800.

[5] X. Cao, G. Cong, and C. S. Jensen, "Mining significant semantic locations from GPS data," *PVLDB*, vol. 3, no. 1, pp. 1009–1020, 2010.

[6] M. R. Uddin, C. V. Ravishankar, and V. J. Tsotras, "Finding regions of interest from trajectory data," in *12th IEEE International Conference on Mobile Data Management, MDM 2011, Volume 1*, 2011, pp. 39–48.

[7] S. Kisilevich, F. Mansmann, and D. A. Keim, "P-DBSCAN: a density based clustering algorithm for exploration and analysis of attractive areas using collections of geo-tagged photos," in *Proceedings of the 1st International Conference and Exhibition on Computing for Geospatial Research & Application, COM.Geo 2010*, 2010.

[8] J. Liu, Z. Huang, L. Chen, H. T. Shen, and Z. Yan, "Discovering areas of interest with geo-tagged images and check-ins," in *Proceedings of the 20th ACM Multimedia Conference, MM '12*, 2012, pp. 589–598.

[9] D. Laptev, A. Tikhonov, P. Serdyukov, and G. Gusev, "Parameter-free discovery and recommendation of areas-of-interest," in *Proceedings of the 22nd ACM SIGSPATIAL International Conference on Advances in Geographic Information Systems*, 2014, pp. 113–122.

[10] G. Cai, C. Hio, L. Bermingham, K. Lee, and I. Lee, "Sequential pattern mining of geo-tagged photos with an arbitrary regions-of-interest detection method," *Expert Syst. Appl.*, vol. 41, no. 7, pp. 3514–3526, 2014.

[11] M. Shirai, M. Hirota, S. Yokoyama, N. Fukuta, and H. Ishikawa, "Discovering multiple hotspots using geo-tagged photographs," in *SIGSPATIAL 2012 International Conference on Advances in Geographic Information Systems*, 2012, pp. 490–493.

[12] Y. Zheng, L. Zhang, Z. Ma, X. Xie, and W. Ma, "Recommending friends and locations based on individual location history," *TWEB*, vol. 5, no. 1, pp. 5:1–5:44, 2011.

[13] C. Parent *et al.*, "Semantic trajectories modeling and analysis," *ACM Comput. Surv.*, vol. 45, no. 4, pp. 42:1–42:32, 2013.

[14] M. D. Choudhury, M. Feldman, S. Amer-Yahia, N. Golbandi, R. Lempel, and C. Yu, "Automatic construction of travel itineraries using social breadcrumbs," in *HT'10, Proceedings of the 21st ACM Conference on Hypertext and Hypermedia*, 2010, pp. 35–44.

[15] X. Lu, C. Wang, J. Yang, Y. Pang, and L. Zhang, "Photo2trip: generating travel routes from geo-tagged photos for trip planning," in *Proceedings of the 18th International Conference on Multimedia 2010*, 2010, pp. 143–152.

[16] K. H. Lim, J. Chan, C. Leckie, and S. Karunasekera, "Personalized tour recommendation based on user interests and points of interest visit durations," in *Proceedings of the Twenty-Fourth International Joint Conference on Artificial Intelligence, IJCAI 2015*, 2015, pp. 1778–1784.

[17] I. R. Brilhante, J. A. F. de Macêdo, F. M. Nardini, R. Perego, and C. Renso, "On planning sightseeing tours with tripbuilder," *Inf. Process. Manage.*, vol. 51, no. 2, pp. 1–15, 2015.

[18] P. Järv, "Extracting human mobility data from geo-tagged photos," in *Proceedings of the PredictGIS'17:1st ACM SIGSPATIAL Workshop on Prediction of Human Mobility*, 2017, pp. 4:1–4:7.

[19] Z. Yan, C. Parent, S. Spaccapietra, and D. Chakraborty, "A hybrid model and computing platform for spatio-semantic trajectories," in *The Semantic Web: Research and Applications, 7th Extended Semantic Web Conference, ESWC 2010, Proceedings, Part I*, 2010, pp. 60–75.

[20] L. O. Alvares, V. Bogorny, B. Kuijpers, J. A. F. de Macêdo, B. Moelans, and A. A. Vaisman, "A model for enriching trajectories with semantic geographical information," in *15th ACM International Symposium on Geographic Information Systems, ACM-GIS 2007, Proceedings*, 2007, p. 22.

[21] D. J. Crandall, L. Backstrom, D. P. Huttenlocher, and J. M. Kleinberg, "Mapping the world's photos," in *Proceedings of the 18th International Conference on World Wide Web, WWW 2009*, 2009, pp. 761–770.

[22] Z. Yin, L. Cao, J. Han, J. Luo, and T. S. Huang, "Diversified trajectory pattern ranking in geo-tagged social media," in *Proceedings of the Eleventh SIAM International Conference on Data Mining, SDM 2011*, 2011, pp. 980–991.

[23] J. Sander, X. Qin, Z. Lu, N. Niu, and A. Kovarsky, "Automatic extraction of clusters from hierarchical clustering representations," in *Advances in Knowledge Discovery and Data Mining, 7th Pacific-Asia Conference, PAKDD 2003, Proceedings*, 2003, pp. 75–87.

[24] R. J. G. B. Campello, D. Moulavi, and J. Sander, "Density-based clustering based on hierarchical density estimates," in *Advances in Knowledge Discovery and Data Mining, 17th Pacific-Asia Conference, PAKDD 2013, Proceedings, Part II*, 2013, pp. 160–172.

[25] D. Luxen and C. Vetter, "Real-time routing with openstreetmap data," in *Proceedings of the 19th ACM SIGSPATIAL International Conference on Advances in Geographic Information Systems*. ACM, 2011, pp. 513–516.

[26] L. McInnes, J. Healy, and S. Astels, "hdbscan: Hierarchical density based clustering," *The Journal of Open Source Software*, vol. 2, no. 11, p. 205, 2017.

[27] H. Edelsbrunner, D. G. Kirkpatrick, and R. Seidel, "On the shape of a set of points in the plane," *IEEE Trans. Information Theory*, vol. 29, no. 4, pp. 551–558, 1983.

[28] S. C. Deerwester, S. T. Dumais, T. K. Landauer, G. W. Furnas, and R. A. Harshman, "Indexing by latent semantic analysis," *JASIS*, vol. 41, no. 6, pp. 391–407, 1990.

[29] W. W. Cohen, P. Ravikumar, and S. E. Fienberg, "A comparison of string distance metrics for name-matching tasks," in *Proceedings of IJCAI-03 Workshop on Information Integration on the Web (IIWeb-03)*, 2003, pp. 73–78.

[30] R. Řehůřek and P. Sojka, "Software Framework for Topic Modelling with Large Corpora," in *Proceedings of the LREC 2010 Workshop on New Challenges for NLP Frameworks*. ELRA, May 2010, pp. 45–50.

[31] Y. Zheng, "Trajectory data mining: An overview," *ACM TIST*, vol. 6, no. 3, pp. 29:1–29:41, 2015.

Appendix 4

Publication IV
P. Järv. Predictability limits in session-based next item recommendation. In T. Bogers, A. Said, P. Brusilovsky, and D. Tikk, editors, *Proceedings of the 13th ACM Conference on Recommender Systems, RecSys 2019, Copenhagen, Denmark, September 16-20, 2019.*, pages 146–150. ACM, 2019

Predictability Limits in Session-based Next Item Recommendation

Priit Järv
Tallinn University of Technology, Estonia
priit@whitedb.org

ABSTRACT

Session-based recommendations are based on the user's recent actions, for example, the items they have viewed during the current browsing session or the sightseeing places they have just visited. Closely related is sequence-aware recommendation, where the choice of the next item should follow from the sequence of previous actions.

We study seven benchmarks for session-based recommendation, covering retail, music and news domains to investigate how accurately user behavior can be predicted from the session histories. We measure the entropy rate of the data and estimate the limit of predictability to be between 44% and 73% in the included datasets.

We establish some algorithm-specific limits on prediction accuracy for Markov chains, association rules and k-nearest neighbors methods. With most of the analyzed methods, the algorithm design limits their performance with sparse training data. The session based k-nearest neighbors are least restricted in comparison and have room for improvement across all of the analyzed datasets.

CCS CONCEPTS

• **Information systems → Recommender systems**; *Test collections*; • **Mathematics of computing** → *Information theory*.

KEYWORDS

Session-based recommendation; Predictability

ACM Reference Format:

Priit Järv. 2019. Predictability Limits in Session-based Next Item Recommendation. In *Thirteenth ACM Conference on Recommender Systems (RecSys '19), September 16–20, 2019, Copenhagen, Denmark.* ACM, New York, NY, USA, 5 pages. https://doi.org/10.1145/3298689.3346990

1 INTRODUCTION

Recommender systems aim to improve the user experience in shops, entertainment or travel apps and targeted advertising, by finding items or activities that the user is interested in. From the business point of view, the user is also more likely to make purchases when the recommender is able to anticipate what the user likes. However, the preferences of the user may not be readily available. The user may interact with the system anonymously or have no previous history. The user may be looking for place recommendations in a country or area they have not previously been to.

From practical perspective, being able to make recommendations in such scenarios is highly relevant. The problem of making recommendations tailored to user sessions, rather than established users, is called session-based recommendation. For the remainder of the paper, we will refer to any sequence of user actions as a *session* and the elements of the sequence as *items*. The items may refer to listened songs, visited sightseeing places or actual items in an online store.

In a related scenario, recommendations may themselves consist of a collection or a sequence of items. Music playlists and trip itineraries are examples where both the individual items and transitions between them are important for the quality of recommendations. Even when the goal is to recommend a single next item, such as the next song to listen, the sequence of previous items is often relevant. Recommender systems that consider the sequence of items are called sequence-aware.

The ability to predict the next item in a sequence correctly is commonly used to evaluate algorithmic approaches in session-based scenarios. While the quality in recommendations goes beyond just being able to find the most likely item, this is still an important indicator in determining whether the model has captured user interests accurately.

Differences in methodology, baselines and evaluation datasets in session-based recommendation research have made comparison and analysis of different algorithmic approaches difficult. To establish a common baseline for evaluating next item predictions, Ludewig and Jannach published a set of benchmark datasets and a standardized methodology [5]. They measured the performance of various methods and concluded that there are no major differences between simple methods like k-nearest neighbors and sophisticated methods like matrix factorization approaches that are intended to overcome the difficulties with simpler methods, such as data sparsity and ability to generalize. In fact, the simpler methods performed better on majority of the datasets.

In this paper, we examine properties of seven public datasets included in the benchmarks of Ludewig and Jannach [5]. To understand what kind of expectations could be placed on the accuracy of future improvements in algorithms, we estimate the predictability of the item sequences in the datasets. We define *predictability* as the probability that the recommender will correctly predict the next item, given an unfinished session and a history of other sessions. Assuming we knew the most likely item each time, the prediction can still be wrong when other items have a non-zero probability of appearing. When discussing the predictability of a dataset, we mean the probability that a perfect predictor will guess correctly, averaged over the dataset.

RecSys '19, September 16–20, 2019, Copenhagen, Denmark

ACM ISBN 978-1-4503-6243-6/19/09…$15.00
https://doi.org/10.1145/3298689.3346990

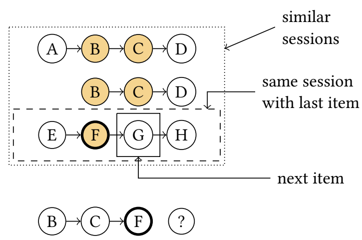

Figure 1: Items available for recommendation. Given a session to recommend for, the algorithms we analyze restrict the possible responses to co-occurrences in the training set with the session items {B, C, F} (marked with color fill).

For simpler algorithms, the design of the algorithm places an upper limit on the ability to predict the next item. For example, when modeling the sessions as a Markov chain, the probability distribution of the next item can only include the items that immediately followed the current item in a session in the training data. The method's accuracy is then limited to how often the known item pairs occur in the testing data, relative to previously unknown item pairings.

Figure 1 illustrates typical cases where the co-occurrence in the training data determines which items are available for recommendation. For a session {B, C, F}, considering only the continuations seen in training data leads to recommending G. Methods that look at entire sessions could also rank E and H. Neighborhood-based methods that find similar sessions can also recommend the item D.

We determine co-occurrence relations that affect the performance of Markov chains, association rules and k-nearest neighbors (k-NN) in all included datasets. These relations are highly dependent on datasets and training-testing splits, but are useful in explaining and further evaluating the performance of the affected methods in the reference benchmark of Ludewig and Jannach [5].

The remainder of the paper is organized as follows. In Section 2 we review the related work. In Section 3 we estimate two types of limits on prediction accuracy - upper limits due to the inherent randomness in the benchmark datasets and limits due to algorithm design. Section 4 summarizes the findings and suggests directions for future research.

2 RELATED WORK

This paper relies extensively on the evaluation of session-based recommenders by Ludewig and Jannach [5]. Recognizing the lack of unified methodology, they collected datasets and performed evaluation of multiple classes of methods. The public datasets and testing framework have also been published online, to facilitate building future research on this platform. They found that with their methodology and an increased variety of datasets there was no significant difference between naive baselines and latest sophisticated approaches. Ludewig and Jannach concluded that more research is required into the circumstances that affect algorithm performance in different datasets. We have made an initial investigation into this question in this paper.

Session-based recommendation can be seen as a subset of sequence-aware recommendation. Quadrana et al. review the applications, types and methods of sequence-aware recommenders [7]. In recent years, the attention has turned to deep learning approaches to recommendation, reviewed by Zhang et al. [19].

The entropy method to estimate predictability of sequential information was popularized by Song et al. They used mobile phone usage logs to calculate the predictability of trajectories of individual mobile phone users [10]. An explanation of the theoretical basis and additional proofs were given by Smith et al. [9]. The details of the entropy estimation of the original method of Song et al. can interpreted in different ways. To address that Xu et al. published a clarified method with experimental evaluation [15].

Substantial prior work exists on making next-item predictions using the datasets included in our evaluation [2–4, 11, 12, 14, 16, 17]. These papers generally use different methodology so we do not compare their results directly to our estimated predictability. They also cover a wider range of recommendation scenarios, including established user histories and leveraging content information [2, 4, 12, 14, 16, 17].

3 LIMITS ON PREDICTION ACCURACY

In this section we determine the limits on prediction accuracy due to inherent randomness in data and due to algorithm design. We consider only the limits on the hit rate (HR) metric, which can be directly estimated with our approach. The hit rate metric measures whether the item to predict was included in the top-n recommended items (abbreviated as HR@n). We cover 7 public datasets included in the benchmark of Ludewig and Jannach[5]:

- *RSC15* - item views and purchases in online retail, published for the ACM RecSys 2015 Challenge[1];
- *TMALL* - online purchasing history from Tmall.com;
- *RETAILR* - user browsing histories from Retail Rocket[8];
- *AOTM* - music playlists from the Art of the Mix platform[6];
- *30MUSIC* - listening histories from last.fm[13];
- *NOWPLAYING* - "currently listening" tweets[18];
- *CLEF* - news article reads by users, a subset of the data used in the 2017 CLEF NewsREEL challenge[5].

We calculate the estimates on training and test splits created with the same methodology and settings as used by Ludewig and Jannach[5] and compare our limit estimations to the performance of algorithms as measured in their benchmark. The programs to reproduce the calculations and instructions to access the data are available online[1].

3.1 Limit on predictability

We use the method of Song et al. [10] to estimate the upper bound on predictability of sequential data through estimating the entropy rate on the sequences. To determine predictability in the benchmark setting, we treat the training and test splits as a single joint sequence. This simplification is safe as long as we do not consider the effects of sessions overlapping or having taken place in parallel important.

[1]https://github.com/priitj/recsys19

Let the next item in a sequence be X_i and the items preceding it h_i. The entropy associated with the next item X_i, as a measurement of how predictable its possible values are, is:

$$H(X_i|h_i) = -\sum_x P(X_i = x|h_i) \log_2 P(X_i = x|h_i) \tag{1}$$

The probability distribution $P(X_i|h_i)$ is the true probability distribution of the next item X_i. While we do not know the true distributions and cannot calculate the quantity $H(X_i|h_i)$ directly, it will form the theoretical basis of determining the maximum predictability of the sequence. The entropy rate of a sequence is[9]:

$$\mathcal{H}(\mathcal{X}) = \lim_{t \to \infty} \frac{1}{t} \sum H(X_t|h_t) \tag{2}$$

Given the entropy rate, the bound on maximum predictability $\Pi^{\overline{max}}$ can be found by numerically solving

$$\mathcal{H}(\mathcal{X}) = -\Pi^{\overline{max}} \log_2 \Pi^{\overline{max}} - (1 - \Pi^{\overline{max}}) \log_2 \frac{1 - \Pi^{\overline{max}}}{m - 1} \tag{3}$$

where m is the number of unique items. The full derivation of Eq. 3 is given by Smith et al. [9]. Finally, we substitute the theoretical $\mathcal{H}(\mathcal{X})$ with the estimate $S \approx \mathcal{H}(\mathcal{X})$ over a sequence of length n:

$$S = \frac{1}{\frac{1}{n} \sum_i \Lambda_i} \log_2 n \tag{4}$$

This is the corrected estimate by Xu et al.[15], where $\Lambda_i = k_{max}^{(i)} + 1$ and $k_{max}^{(i)}$ is defined as the length of longest sub-sequence starting from position i that appears as a continuous sub-sequence between positions $1 \dots i-1$. Because we use an estimate of the entropy rate, the calculated predictability should be treated as an approximation. Furthermore, the experiments of Xu et al. show that while the corrected method we use is more accurate, it can also underestimate predictability [15].

We include training and test sets because we are interested in determining the entropy rate of the same stochastic process that "produced" the sessions in both sets. The sets are represented by arranging the sessions sequentially and introducing a session end marker e, which is placed between individual sessions. The appearance of the marker in sequence is then the event that the session ends, given all the previous sessions and the content of the current session, with probability $P(X_i = e|h_i)$. Similarly, the probability that the next session starts with item x, given all the previous sessions, is $P(X_i = x|e, h_{i-1})$. Therefore, the sequence can still be viewed as events produced by a stochastic process and Equations 1–2 apply.

We report the average $\Pi^{\overline{max}}$ over five evaluation splits for each dataset in Table 1. The estimated predictability $\Pi^{\overline{max}}$ is a limit on the HR@1 accuracy metric, the ratio of tests where the recommender is able to recommend the correct item as first in a ranked list. The limit is between 44% and 73%, depending on the dataset. Purely session-based recommendation algorithms should not be expected to improve above these values. The state of the art results (column *best known*) are significantly below the limit.

The practical implications of this result depend on the application. In many cases, offering more than one recommendation is useful, so HR@1 is a too strict measure and performance with $n > 1$ is more informative. On the other hand, where the exact prediction or the top position in a ranked list of recommendations is important, the performance will be directly bound by these limits. Finally, the tools we have developed allow calculating predictability estimates on other datasets.

Table 1: The estimate of predictability from entropy. $\Pi^{\overline{max}}$ is the upper limit on the HR@1 metric. *Best known* – HR@1 benchmark performance of the best algorithm on the same data [5].

	$\Pi^{\overline{max}}$	best known[5] HR@1
RSC15	0.65	0.18
TMALL	0.58	0.13
RETAILR	0.59	0.27
AOTM	0.44	0.0096
30MUSIC	0.73	0.20
NOWPLAYING	0.71	0.076
CLEF	0.64	0.12

3.2 Limits due to algorithm design

In this section we evaluate the algorithmic limits of the following methods:

- *MC* - Markov Chain, where the transition probabilities are direct statistical probabilities learned from item-to-item transitions in training data;
- *SR* - Sequential Rules: like Markov Chain, but items indirectly following the current item are included as possible transitions with a decaying weight;
- *AR* - Association Rules: transition probabilities are learned statistically by counting item co-occurrences in training sessions, regardless of sequence;
- *IKNN* - Item-based k-NN, where nearest neighbors are found by comparing item vectors;
- *SKNN* - A family of session-based k-NN, where the common feature is that the nearest neighbors are found by comparing session vectors.

We use the method definitions of Ludewig and Jannach. Further details on the specific implementation choices can be found in their paper [5]. We do not include matrix factorization and neural network methods. While similar analysis of their performance would be highly relevant, it would require a more sophisticated mathematical approach than we have taken here.

With several of the analyzed methods, direct statistical learning from the training data is applied. For example, if the training set included sessions $\{A, B\}$, $\{B, A\}$ and $\{C, A\}$ then with the association rule method we would predict the item B after A with the probability $\frac{2}{3}$ and item C with probability $\frac{1}{3}$. The method would not be able to predict any other items as it has no grounds to do so based on the training data.

With simple recommendation approaches we can directly determine the limit of prediction accuracy since the algorithm design itself restricts which items can be recommended. By determining how the algorithm finds the items to recommend, we count the test instances where it is impossible to give the correct prediction because the required association has not occurred in the training data. The ratio of such cases gives us an exact upper bound on the HR@*n* metric. The limit applies for any *n*. We give the results by algorithm, averaged over five training and test splits, in Table 2.

These results apply to the testing scenario where the recommender is given an incomplete session and has to predict the next item in the session correctly. In the following analysis, we refer to the incomplete session as the *current session*, and the last known item as the *current item*. The item that the recommender is expected to find, is referred to as the *next item*.

Figure 2 shows the limits of prediction accuracy for the most restricted algorithms, compared to their HR@20 performance and the best HR@20 result (state of the art), as measured by [5]. For the MC method, the limit is the ratio of test cases where the next item occurred directly after the current item in training data. The SR and AR methods can recommend the next item if there is a session in training data where the next item occurs somewhere after the current item, or anywhere in the session, respectively.

Overall, Figure 2 suggests that these three algorithms have similar dataset-dependent behaviour. The limits are higher in the *RSC15* and especially the *CLEF* dataset. On the e-commerce and music datasets they are low enough to directly influence the performance. On the *RETAILR* dataset, all three algorithms have a hard limit on the HR@*n* performance that has already been exceeded by the best known benchmark result at HR@20. With SR, this extends to the *AOTM* and *30MUSIC* datasets and with MC to all e-commerce and music datasets.

The significance of the limits decreases when more training data is available. The *CLEF* and *RSC15* datasets have on the average 2200 and 48 sessions per unique item in the training splits used, making them relatively least sparse in terms of training samples per item. The music datasets are the most sparse, having 0.36 or fewer sessions per unique item.

k-NN methods fall under two different categories. The IKNN method finds the items to recommend by calculating the cosine similarity of binary item vectors where the elements correspond to sessions. If the next item has not occurred in the same session with the current item, the dot product of the vectors will be 0. The IKNN method therefore has the same hard limit on the performance in HR@*n* metrics as AR.

In contrast with the other analyzed methods, the family of SKNN methods is much less restricted by design. They find the neighborhood by comparing session vectors, where the elements correspond to items. Whenever the next item has co-occurred with any of the items in the current session, it can potentially be recommended. In the datasets we analyzed, this covers over 80% of test cases each time and in particular, 92% for *AOTM* where other approaches are severely restricted.

Table 2: Limits on prediction accuracy (HR@*n*) for simpler approaches.

	MC	SR	AR, IKNN	SKNN
RSC15	0.79	0.86	0.90	0.97
TMALL	0.25	0.40	0.45	0.90
RETAILR	0.36	0.44	0.57	0.80
AOTM	0.015	0.042	0.079	0.92
30MUSIC	0.29	0.36	0.48	0.91
NOWPLAYING	0.17	0.28	0.40	0.90
CLEF	0.96	0.98	0.99	0.995

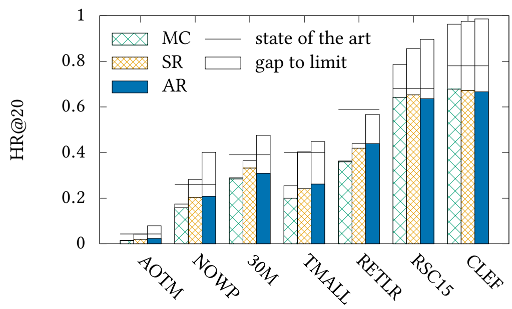

Figure 2: Performance (*HR@20* metric, colored bars) and remaining gap to limit due to algorithm design (clear bars). *State of the art* is the top result for the dataset (any algorithm). The performance measurements were taken from [5].

4 CONCLUSIONS AND FUTURE WORK

We gave estimates between 44% and 73% on the predictability of session-based recommendation scenarios with the analyzed datasets. The accuracy of the exact prediction of the next user action is limited by these values.

Algorithm-specific limits have a practical effect on the performance of the Markov chain, sequential rule, association rule and IKNN methods. While these methods were found to perform well in an earlier study [5], in several cases they cannot be improved significantly due to self-imposed restrictions. With sparse training data, their maximum theoretical performance is already below of what has been demonstrated in practice by other algorithms in the analyzed benchmark.

Session-based *k*-NN methods have room for improvement in all of the evaluated datasets. As they are already competitive with the more sophisticated approaches, more effort should be devoted into developing the SKNN algorithm family.

The methods presented in this paper have several limitations that should be addressed in future work. The estimates of predictability for other metrics that consider multiple recommendations and ranking positions, like the mean reciprocal rank (MRR) would be of practical value. The algorithm-specific analysis should be extended to important sequential recommendation methods – matrix factorization and neural networks.

ACKNOWLEDGMENTS

The authors would like to thank the anonymous referees for their valuable comments and helpful suggestions.

REFERENCES

[1] David Ben-Shimon, Alexander Tsikinovsky, Michael Friedmann, Bracha Shapira, Lior Rokach, and Johannes Hoerle. 2015. RecSys Challenge 2015 and the YOOCHOOSE Dataset. In *Proceedings of the 9th ACM Conference on Recommender Systems, RecSys 2015, Vienna, Austria, September 16-20, 2015*, Hannes Werthner, Markus Zanker, Jennifer Golbeck, and Giovanni Semeraro (Eds.). ACM, 357–358. https://dl.acm.org/citation.cfm?id=2798723

[2] Negar Hariri, Bamshad Mobasher, and Robin D. Burke. 2012. Context-aware music recommendation based on latenttopic sequential patterns. In *Sixth ACM Conference on Recommender Systems, RecSys '12, Dublin, Ireland, September 9-13, 2012*, Padraig Cunningham, Neil J. Hurley, Ido Guy, and Sarabjot Singh Anand (Eds.). ACM, 131–138. https://dl.acm.org/citation.cfm?id=2365979

[3] Balázs Hidasi and Alexandros Karatzoglou. 2018. Recurrent Neural Networks with Top-k Gains for Session-based Recommendations. In *Proceedings of the 27th ACM International Conference on Information and Knowledge Management, CIKM 2018, Torino, Italy, October 22-26, 2018*, Alfredo Cuzzocrea, James Allan, Norman W. Paton, Divesh Srivastava, Rakesh Agrawal, Andrei Z. Broder, Mohammed J. Zaki, K. Selçuk Candan, Alexandros Labrinidis, Assaf Schuster, and Haixun Wang (Eds.). ACM, 843–852. https://doi.org/10.1145/3269206.3271761

[4] Dietmar Jannach, Iman Kamehkhosh, and Lukas Lerche. 2017. Leveraging multidimensional user models for personalized next-track music recommendation. In *Proceedings of the Symposium on Applied Computing, SAC 2017, Marrakech, Morocco, April 3-7, 2017*, Ahmed Seffah, Birgit Penzenstadler, Carina Alves, and Xin Peng (Eds.). ACM, 1635–1642. https://doi.org/10.1145/3019612.3019756

[5] Malte Ludewig and Dietmar Jannach. 2018. Evaluation of session-based recommendation algorithms. *User Model. User-Adapt. Interact.* 28, 4-5 (2018), 331–390. https://doi.org/10.1007/s11257-018-9209-6

[6] Brian McFee and Gert R. G. Lanckriet. 2012. Hypergraph Models of Playlist Dialects. In *Proceedings of the 13th International Society for Music Information Retrieval Conference, ISMIR 2012, Mosteiro S.Bento Da Vitória, Porto, Portugal, October 8-12, 2012*, Fabien Gouyon, Perfecto Herrera, Luis Gustavo Martins, and Meinard Müller (Eds.). FEUP Edições, 343–348. http://ismir2012.ismir.net/event/papers/343-ismir-2012.pdf

[7] Massimo Quadrana, Paolo Cremonesi, and Dietmar Jannach. 2018. Sequence-Aware Recommender Systems. *ACM Comput. Surv.* 51, 4 (2018), 66:1–66:36. https://doi.org/10.1145/3190616

[8] Retailrocket. 2017. Retailrocket recommender system dataset. https://www.kaggle.com/retailrocket/ecommerce-dataset.

[9] Gavin Smith, Romain Wieser, James Goulding, and Duncan Barrack. 2014. A refined limit on the predictability of human mobility. In *IEEE International Conference on Pervasive Computing and Communications, PerCom 2014, Budapest, Hungary, March 24-28, 2014*. IEEE Computer Society, 88–94. https://doi.org/10.1109/PerCom.2014.6813948

[10] Chaoming Song, Zehui Qu, Nicholas Blumm, and Albert-László Barabási. 2010. Limits of predictability in human mobility. *Science* 327, 5968 (2010), 1018–1021.

[11] Yong Kiam Tan, Xinxing Xu, and Yong Liu. 2016. Improved Recurrent Neural Networks for Session-based Recommendations. In *Proceedings of the 1st Workshop on Deep Learning for Recommender Systems, DLRS@RecSys 2016, Boston, MA, USA, September 15, 2016*, Alexandros Karatzoglou, Balázs Hidasi, Domonkos Tikk, Oren Sar Shalom, Haggai Roitman, Bracha Shapira, and Lior Rokach (Eds.). ACM, 17–22. https://doi.org/10.1145/2988450.2988452

[12] Roberto Turrin, Andrea Condorelli, Paolo Cremonesi, Roberto Pagano, and Massimo Quadrana. 2015. Large scale music recommendation. In *Workshop on Large-Scale Recommender Systems (LSRS 2015) at ACM RecSys*.

[13] Roberto Turrin, Massimo Quadrana, Andrea Condorelli, Roberto Pagano, and Paolo Cremonesi. 2015. 30Music Listening and Playlists Dataset. In *Poster Proceedings of the 9th ACM Conference on Recommender Systems, RecSys 2015, Vienna, Austria, September 16, 2015. (CEUR Workshop Proceedings)*, Pablo Castells (Ed.), Vol. 1441. CEUR-WS.org. http://ceur-ws.org/Vol-1441/recsys2015_poster13.pdf

[14] Pengfei Wang, Jiafeng Guo, Yanyan Lan, Jun Xu, Shengxian Wan, and Xueqi Cheng. 2015. Learning Hierarchical Representation Model for NextBasket Recommendation. In *Proceedings of the 38th International ACM SIGIR Conference on Research and Development in Information Retrieval, Santiago, Chile, August 9-13, 2015*, Ricardo A. Baeza-Yates, Mounia Lalmas, Alistair Moffat, and Berthier A. Ribeiro-Neto (Eds.). ACM, 403–412. https://doi.org/10.1145/2766462.2767694

[15] Paiheng Xu, Likang Yin, Zhongtao Yue, and Tao Zhou. 2019. On predictability of time series. *Physica A: Statistical Mechanics and its Applications* 523 (2019), 345–351.

[16] Haochao Ying, Fuzhen Zhuang, Fuzheng Zhang, Yanchi Liu, Guandong Xu, Xing Xie, Hui Xiong, and Jian Wu. 2018. Sequential Recommender System based on Hierarchical Attention Networks. In *Proceedings of the Twenty-Seventh International Joint Conference on Artificial Intelligence, IJCAI 2018, July 13-19, 2018, Stockholm, Sweden.*, Jérôme Lang (Ed.). ijcai.org, 3926–3932. https://doi.org/10.24963/ijcai.2018/546

[17] Feng Yu, Qiang Liu, Shu Wu, Liang Wang, and Tieniu Tan. 2016. A Dynamic Recurrent Model for Next Basket Recommendation. In *Proceedings of the 39th International ACM SIGIR conference on Research and Development in Information Retrieval, SIGIR 2016, Pisa, Italy, July 17-21, 2016*, Raffaele Perego, Fabrizio Sebastiani, Javed A. Aslam, Ian Ruthven, and Justin Zobel (Eds.). ACM, 729–732. https://doi.org/10.1145/2911451.2914683

[18] Eva Zangerle, Martin Pichl, Wolfgang Gassler, and Günther Specht. 2014. #nowplaying Music Dataset: Extracting Listening Behavior from Twitter. In *Proceedings of the First International Workshop on Internet-Scale Multimedia Management, WISMM '14, Orlando, Florida, USA, November 7, 2014*, Roger Zimmermann and Yi Yu (Eds.). ACM, 21–26. https://doi.org/10.1145/2661714.2661719

[19] Shuai Zhang, Lina Yao, Aixin Sun, and Yi Tay. 2019. Deep Learning Based Recommender System: A Survey and New Perspectives. *ACM Comput. Surv.* 52, 1 (2019), 5:1–5:38. https://dl.acm.org/citation.cfm?id=3285029

Curriculum Vitae

1. Personal data

Name	Priit Järv
Date and place of birth	April 4th, 1978, Tallinn, Estonia
Nationality	Estonian

2. Contact information

E-mail	priit.jarv1@taltech.ee

3. Education

2015–2020	Tallinn University of Technology, Information and Communication Technology, PhD studies
2011–2015	Tallinn University of Technology, Informatics, MSc *cum laude*
1995–2011	Tallinn University of Technology, Computer and Systems Engineering, BSc

4. Language competence

Estonian	native
English	fluent
Japanese	conversational
Finnish	conversational

5. Professional employment

2017–2020	Tallinn University of Technology, Early Stage Researcher
2016–2016	Tallinn University of Technology, Visiting Lecturer
2015–2018	Tallinn University, Visiting Lecturer
2009–2015	ELIKO Competence Centre, Subcontractor
2001–2009	Tieto Eesti AS, System Analyst, System Architect, Product Manager
1999–2001	Estonian Citizenship and Migration Board, Systems Administrator
1997–1998	Tallinn University of Technology, Systems Administrator
1997–1997	Estonian Citizenship and Migration Board, Programmer

Elulookirjeldus

1. Isikuandmed

Nimi	Priit Järv
Sünniaeg ja -koht	04.04.1978, Tallinn, Eesti
Kodakondsus	Eesti

2. Kontaktandmed

E-post	priit.jarv1@taltech.ee

3. Haridus

2015–2020	Tallinna Tehnikaülikool, Info- ja kommunikatsioonitehnoloogia, doktoriõpe
2011–2015	Tallinna Tehnikaülikool, Informaatika, MSc *cum laude*
1995–2011	Tallinna Tehnikaülikool, Arvutisüsteemid, BSc

4. Keelteoskus

eesti keel	emakeel
inglise keel	kõrgtase
jaapani keel	kesktase
soome keel	kesktase

5. Teenistuskäik

2017–2020	Tallinna Tehnikaülikool, doktorant-nooremteadur
2016–2016	Tallinna Tehnikaülikool, külalislektor
2015–2018	Tallinna Ülikool, külalislektor
2009–2015	ELIKO Tehnoloogia Arenduskeskus OÜ, alltöövõtja
2001–2009	Tieto Eesti AS, analüütik, süsteemiarhitekt, tootejuht
1999–2001	Kodakondsus- ja Migratsiooniamet, IT süsteemide administraator
1997–1998	Tallinna Tehnikaülikool, IT süsteemide administraator
1997–1997	Kodakondsus- ja Migratsiooniamet, programmeerija