Semantic Classification of Locations in Human Trajectories Using Hidden Markov Models

Rúben da Costa Raposo

Thesis to obtain the Master of Science Degree in:
Information Systems and Computer Engineering

Supervisor: Prof. Doutor Bruno Emanuel da Graça Martins

Examination Committee
Chairperson: Prof. Doutor João Emílio Segurado Pavão Martins
Supervisor: Prof. Doutor Bruno Emanuel da Graça Martins
Member of the Committee: Prof.ª Doutora Ana Paula Afonso

June 2015
Abstract

With the increasing availability of geopositioning information, due to the massification of devices combining GPS receivers with the access to location-based services and social networks, we are also witnessing a growing interest in the analysis of human location histories. In my MSc thesis, I compared different approaches for the classification of visited locations within human trajectories, according to semantic categories. I specifically experimented with an approach based on heuristics, and with two other approaches based on Hidden Markov Models (HMMs), relying on either a supervised or an unsupervised training setting. HMMs can take into account the location characteristics as unobservable context, relating this information beneath a time process, in our case corresponding to the trajectories. For conducting a series of experiments with the aforementioned methods, I built a dataset that combines the previously available GeoLife trajectory data with information collected from the Foursquare location-based social network. Results show that a classification accuracy of 56.5% can be achieved with the supervised HMM, when considering discrete observations that correspond to regions approximately with areas from 740 to 1534m², in the modeling of the trajectories.

Keywords: Hidden Markov Models, Semantic Location Classification, Human Mobility Analysis, Trajectory Mining, Machine Learning
Resumo

Com o aparecimento de cada vez mais informação georreferenciada, devido à massificação da utilização de aparelhos que combinam receptores GPS com o acesso a serviços e redes sociais baseados em localização, assistimos a um crescente interesse na análise de históricos de localização referentes à mobilidade de atores humanos. Na minha tese de mestrado, comparei diferentes abordagens para a classificação de locais visitados em trajetórias humanas. Mais especificamente, as abordagens que utilizei para classificar os locais visitados através de categorias semânticas são baseadas em heurísticas e em modelos de Markov com variáveis ocultas (HMMs), sendo que estes últimos foram construídos com abordagens que consideram diferentes tipos de aprendizagem, nomeadamente supervisionada ou não supervisionada. Os modelos HMM consideram as características dos locais visitados como contexto não observável e relacionam esta informação num processo sequencial, que neste caso corresponde às trajetórias. Para a condução de uma série de experiências, construi um conjunto de dados que combina a informação previamente disponibilizada pelo conjunto de trajetórias GeoLife, com informação recolhida na rede social Foursquare. Os resultados mostram que uma eficácia de 56,5% na classificação de locais pode ser atingida ao utilizar um HMM supervisionado, quando consideradas observações para o HMM correspondendo a regiões com áreas compreendidas entre 740m² e 1534m², para a modelação dos locais existentes nas trajetórias.

Keywords: Modelos Hidden Markov, Classificação Semântica de Locais, Análise de Mobilidade Humana, Significado associado a Trajetórias, Aprendizagem Automática
Acknowledgements

I want to start by thanking my parents for the unconditional support, patience and motivation throughout these last years and, specially, during my MSc thesis period.

I have a lot to thank my advisor for the constant support, comprehension and availability in these months of work. The conclusion of this document would have been harder without his regular feedback. I also want to thank him for the opportunity to participate on the Services for Intelligent Geographical Information Systems (SInteliGIS) project founded by the Fundação para a Ciência e a Tecnologia (FCT), through the project grant with reference PTDC/EIA-EIA/109840/2009.

To my friends, I thank the support and the good moments during these stressful times.

Finally, I would like to leave a special thanks to my girlfriend for all the love, partnership, adventures in our academic years and, above all, for her patience during these past few months.
# Contents

Abstract  
Resumo  
Acknowledgements  

## 1 Introduction

1.1 Hypothesis and Methodology  
1.2 Contributions  
1.3 Thesis Structure  

## 2 Concepts and Related Work

2.1 Fundamental Concepts  
2.1.1 Models for Sequential Data Classification  
2.2 Related Work  
2.2.1 Data Mining Techniques  
2.2.2 Sequence Classification Approaches  
2.2.3 Hybrid Approaches  
2.3 Summary  

## 3 Semantic Classification of Locations within Trajectories  
3.1 Overview  
3.2 Heuristic Trajectory Classification  

1  
2  
4  
5  
7  
7  
10  
15  
15  
16  
22  
27  
29  
29  
30  

v  
iii  
v
<table>
<thead>
<tr>
<th>Section</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>3.3 Human Trajectory Classification by Using Hidden Markov Models</td>
<td>32</td>
</tr>
<tr>
<td>3.3.1 Handling Spatio-Temporal Positions Through a Triangular Mesh</td>
<td>33</td>
</tr>
<tr>
<td>3.3.2 Using HMMs in a Supervised Setting</td>
<td>35</td>
</tr>
<tr>
<td>3.3.3 Using HMMs in an Unsupervised Setting</td>
<td>36</td>
</tr>
<tr>
<td>3.4 Summary</td>
<td>39</td>
</tr>
<tr>
<td>4 Validation Experiments</td>
<td>41</td>
</tr>
<tr>
<td>4.1 The GeoLife Dataset</td>
<td>41</td>
</tr>
<tr>
<td>4.2 Evaluation Methodology</td>
<td>44</td>
</tr>
<tr>
<td>4.3 Experimental Results</td>
<td>46</td>
</tr>
<tr>
<td>4.3.1 Results with Supervised HMMs</td>
<td>46</td>
</tr>
<tr>
<td>4.3.2 Results with Unsupervised HMMs</td>
<td>52</td>
</tr>
<tr>
<td>4.4 Summary</td>
<td>55</td>
</tr>
<tr>
<td>5 Conclusions</td>
<td>57</td>
</tr>
<tr>
<td>5.1 Summary of Results</td>
<td>58</td>
</tr>
<tr>
<td>5.2 Future Work</td>
<td>59</td>
</tr>
<tr>
<td>Bibliography</td>
<td>61</td>
</tr>
</tbody>
</table>
List of Tables

2.1 Candidate path sets and the respective geographic scores (adapted from Ying et al. (2011)). .......................................................... 26

3.2 Category keywords. ........................................................... 32

4.3 Dataset characterization considering different geospatial resolutions. ................. 43

4.4 Results of different supervised experiments. ................................................. 46

4.5 Results for each of the individual HMM states. ............................................. 47

4.6 The training and testing datasets used in the case of model 18_{10}. .................. 50

4.7 Results of different unsupervised experiments. ............................................. 54

4.8 Confusion matrix of the supervised model 18_{10} concerning the visit classifications (i.e., the proposed states) and their categories. ................................. 55
## List of Figures

2.1 Execution of the Viterbi algorithm for the sequence $AABC$.

2.2 Illustration for some of the limitations of first-order Hidden Markov Models.

2.3 Mixed-Markov-chain model (adapted from Asahara et al. (2011)).

2.4 The concept hierarchy for location-based activity recognition (adapted from Liao et al. (2007)).

2.5 CRF for associating GPS measurements to street patches (adapted from Liao et al. (2007)). The different types of cliques are shown in shaded areas.

2.6 The *SemanPredict* framework for location prediction (adapted from Ying et al. (2011)).

2.7 An example of cell semantic trajectories (adapted from Ying et al. (2011)).

2.8 The *SLP-tree* and the score of geographic behavior (adapted from Ying et al. (2011)).

3.9 Recursive triangular division of a sphere through a hierarchical triangular mesh.

3.10 The modeling of each trajectory for the purpose of training a supervised HMM.

4.11 Example for location windows used in the trajectory filtering process.

4.12 Location distribution by category present in the dataset, with a geospatial resolution of 18.

4.13 Probabilities associated to the supervised model $18_{10}$.

4.14 Number of visits per hour for each category.

4.15 Number of visits per day for each category.
Chapter 1

Introduction

In the present social era, the massification of devices capable of using the global positioning system (GPS) results in a new and huge interest on the analysis of human location histories. Such interest around human location histories is continuously growing due to the increasing availability of GPS generated data, which reflects our daily behavior. In addition, and attracting an even greater attention to this issue, we have that the vast majority of these GPS capable devices offer the opportunity to share, on social networks, our activities and experiences at a given location. A particular type of these networks, namely location-based social networks (LBSN), focuses on exposing information about locations, from basic data to users’ more detailed and individual feedback (e.g., comments, ratings or categorizations for a particular local business). The most famous LBSNs are Foursquare\(^1\), Yelp\(^2\), Facebook Places\(^3\) and Whrrl\(^4\).

Nowadays, the scientific community frequently studies human location histories, where the state-of-the-art work includes sophisticated techniques for data modeling, storage, indexing, querying and mining of human mobility patterns (Zheng & Zhou, 2011). Nonetheless, the work done so far mainly focuses on spatial-temporal features of location histories, leaving the opportunity to explore high-level semantic features. The knowledge extraction from semantic features, through the identification of new behavioral information (e.g., patterns, events and habits), essentially represents the possibility to improve the accuracy on classification or prediction tasks and an extension to the scenarios where mobile data can be applied (e.g., health care or leisure).

\(^1\)http://www.foursquare.com/
\(^2\)http://www.yelp.com
\(^3\)https://www.facebook.com/places/
\(^4\)http://www.whrrl.com/
In my MSc thesis, I address the problem of semantic classification of locations in human trajectories. Specifically, I built and compared different approaches to semantically classify the visited locations within human trajectories, one of them based on heuristics and two others based on Hidden Markov Models (HMMs). The HMM-based approaches were developed by considering both supervised and unsupervised settings. I performed a large set of experiments using the Microsoft GeoLife dataset, enriched with semantic data gathered from Foursquare, specifically the available main categories (e.g., Shopping or Nightlife).

The major contributions of this work are the proposed approaches to semantically classify visited locations within a trajectory. To find the context of the visited locations in a given trajectory, I use a set of spatio-temporal features, namely latitude, longitude and timestamp derived features. With these approaches and considering 9 possible categories to classify each location, I obtained an accuracy of 56.5% for the supervised setting and 33.5% for the unsupervised one.

1.1 Hypothesis and Methodology

In my MSc thesis, I studied the problem of semantic classification of location in human trajectories. More specifically, I conducted experiments to validate hypothesis which states that one can effectively apply sequential models in semantic classification of locations within trajectories. Sequential models are adequate for this particular classification task, since they can represent the existing spatio-temporal relations between the locations within a given trajectory.

In the experiments addressing the previously described semantic classification task, the HMM models were implemented considering distinct training and normalization approaches. In brief, HMMs can take into account the location characteristics as unobservable context, relating this information beneath a time process, in our case corresponding to trajectories. Briefly, an HMM is a generative probabilistic model that explains a process, with a given duration $T$, containing symbols $x_t$ (e.g., representing locations) that are directly influenced by a corresponding state $y_t$ (i.e., encapsulating the unobserved context), which in turn is also influenced by the previous state $y_{t-1}$. The HMM model defines the generation of sequences of symbols through three contributing factors. The first is the probability of the sequence starting at a particular state. The second is the probability of choosing a given state after another (i.e., transition probabilities). The latter is the probability of the production of the different symbols in each state (i.e., emission probabilities).

In order to validate the research hypothesis, the GeoLife dataset was initially enriched with semantic classes for each of the visited locations. The data enrichment process was made using a semi-automatic approach that corresponds to the first experimental method, which leveraged on
1.1. HYPOTHESIS AND METHODOLOGY

data collected from a location based social network, namely, Foursquare. The enriched dataset was then used as the basis for the experiments with the HMM approaches. In the experiments, both supervised and unsupervised settings were considered for the training of HMMs.

For assessing the performance of the different approaches that were proposed, a set of evaluation metrics was designed and implemented considering the type of input (i.e., trajectories) to be used with HMM models. The designed evaluation metrics were intended to clearly show the performance of the experimented models from different perspectives, namely presenting the results for the locations regardless of the trajectories where they are included, or by considering the trajectories as a whole and each category independently.

In the case of supervised approaches, the semantic locations’ classes (i.e., categories) obtained through the semi-automatically approach were used to represent the states of the HMM model. Therefore, when calculating the performance metrics, for each tested location it was possible to directly compare its semi-automatic obtained class with the HMM’s proposed state.

Regarding the unsupervised approaches, the models consider a pre-determined amount of generic states since, during the learning process, the categories obtained by the semi-automatic process to enrich the dataset’s locations were not accountable. Since it was interesting to compare the classification results across all the experimented models, it was necessary to find a method that could relate the states considered in both the supervised and unsupervised models.

Considering this requirement, after any classification process, each tested location has associated a predicted state by the model and a category attributed by the heuristic that previously enriched the dataset (i.e., two distributions associating each category and each state to a group of locations). It is possible to compare models from both learning approaches using the categories attributed to the locations. Therefore, a possible solution was to determine a map of generic state-category pairs for the unsupervised models and then compare the results of each one of them to the results obtained for the corresponding supervised state. Note that the designations of the categories were used to represent the states in the supervised models.

The idea of the method, to determine the map of state-category pairs, was to compare the group of locations associated with each category to the ones associated with each state, in order to find which group of locations are more similar. For comparing the groups of locations, the method applied the mean square root deviation measure using time-related information about each location composing them. After the calculus of the differences between all the groups of locations associated with categories to the ones associated with states, the state-category pairs were formed through the observation of the smaller differences, without repeating pairs of elements.
CHAPTER 1. INTRODUCTION

1.2 Contributions

The main contributions presented in my MSc dissertation are as follows:

- The main contribution of this dissertation was the validation of the ability of sequential models (i.e., HMMs) to perform the semantic classification of the visited locations within human trajectories. The validation was executed through a series of experiments whose results achieved an accuracy of 56% for the models trained using supervised learning and 34% for the models whose learning approach is an unsupervised one. Also in the context of the obtained results and experiments, a smoothing strategy was applied to the HMM models without changing its structure, showing that it is possible to meaningfully improve the results. More specifically, when the smoothing strategy is applied, I observed an accuracy increase of 13.2% in the best supervised model and a global maximum of 15.6% in another supervised model considering smaller areas. The smoothing strategy offers a solution for the sparse data problem of the considered and available locations in the dataset;

- The study and comparison of distinct approaches concerning the semantic classification of locations in human trajectories, with a special focus on HMMs. For training the models, a standard supervised algorithm and a variant including a novel smoothing technique were implemented. Regarding the unsupervised training, the Baum-Welch algorithm was adapted to include the same novel smoothing technique. Also, to enable the assessment of the quality achieved by the unsupervised model, a custom algorithm was built to map the HMM states to semantic categories, using the root mean square deviation measure combined with timestamp features (i.e., hour of the day and day of the week). All the previous code was made available as an open-source project 1, in order to be used by other researchers or persons interested in this topic;

- For enriching the GeoLife2 trajectory dataset, a procedure was implemented to collect data from Foursquare. The information within GeoLife is associated with JSON objects including semantic information about each location. In the context of this work, only the Foursquare’s primary categories associated with each location were considered. Using as input the primary categories, an algorithm was implemented to filter the data and assign only a main category to each location. Again, and to be at the disposal of the research community, all the previously mentioned code was made available as an open-source project1;

---

1 https://code.google.com/p/alochist/
2 http://research.microsoft.com/en-us/downloads/b16d359d-d164-469e-9fd4-daa38f2b2e13/
1.3 Thesis Structure

The remaining content of this document is organized as follows. Chapter 2 introduces the fundamental concepts involved in the semantic classification of locations, presenting models that can address sequential data classification, and presenting the most relevant related work. Chapter 3 describes the designed feature sets and details the several classifiers, built by the combination of models and feature sets, tested in the context of this work. Chapter 4 describes the validation methodology, presents the results that were achieved in the experiments, and presents a critical discussion. Finally, Chapter 5 presents the main conclusions of this work, also discussing the possible paths for future work.
Chapter 2

Concepts and Related Work

This chapter introduces the main concepts, required for understanding the proposed ideas, and presents the most significant related work, with the aim of providing the reader a clear understanding of the task of semantic classification of locations in human trajectories. After a brief introduction to the elementary concepts related to classification tasks using georeferenced information, the distinct models for sequential data classification are then presented. The chapter ends by presenting the most relevant related work, which helps to understand both the problems associated with trajectory classification and the practical application of previously proposed sequence classification models.

2.1 Fundamental Concepts

A clear definition of the elementary concepts is essential to understand the work done in the context of my MSc thesis. Thus, this section aims to synthesize these concepts and provide a high-level description about the information that composes them.

- **Sequence or Trajectory** is an ordered set of visited locations collected by a GPS receiver, within a time frame. Formally, a trajectory is represented as a sequence \( t = < l_0, ..., l_n > \), in which \( n \) is the number of locations and where the \( l_n \) is the last location in the sequence. Note that the number of locations may differ from one trajectory to another, but we always respect the constraint where at least two locations compose a trajectory (i.e., \( n > 1 \)).

- **Visit** is a location enriched with contextual information. This contextual information describes the user activity at a specific location.
• **Location** identifies a point or a region on the Earth's surface. Considering raw data as available in the GeoLife dataset, a location is characterized by the latitude, longitude and timestamp attributes.

• **Index codes** translate coordinates (i.e., latitude and longitude) into a space of discrete values. These codes have the purpose of reducing the size of the sample of locations while maintaining its fundamental properties. The Hierarchical Triangular Mesh method was used to index locations, since it preserves the topology of a spherical form (i.e., Earth surface) and allowed to efficiently experiment with HMM models considering areas with different sizes (Szalay et al., 2007). For further details and examples see Section 3.3.1.

• **Symbol** is a data representation of the observations in sequence classification models. Particularly, in the context of this task, a symbol is a spatio-temporal representation of the visited locations within trajectories that are used as observations in the sequential models. One such model uses symbols in the classification process to determine the context of the visit. If the symbols (i.e., the information included in them) correlate and describe well the proposed (or a given) classes, then the learning and classification processes are easier. The knowledge and perception about the task, the understanding of its properties and how they will leverage the model (i.e., considering its strengths and limitations) are of critical importance to define the symbols and for the success of the classification. Taking into account that visits are elements composing trajectories, and the available raw data, spatial and temporal information were used to define the symbols of the proposed task, namely the index code, day of the week and hour of the day. More specifically, and considering that symbols are discrete observations, the index code is a discrete representation of the original continuous coordinates through the Hierarchical Triangular Mesh. The day of the week and the hour of the day are equally discrete representations of the visited location’s timestamp.

• **Semantic Label** is a description for identifying the user’s activity at a specific location. For enriching each of the locations in the GeoLife trajectories with contextual information, a heuristic was used to filter the semantic categories from Foursquare associated with a given location. The heuristic and some related challenges are presented in Section 3.2.

• **Model** is an object built through an algorithm that learns from data (i.e., from a trajectory set). Later, predictions and decisions are made by the model, instead of following explicit programmed instructions. In my MSc thesis, the dataset containing the trajectories is randomly divided into two distinct sets. Then, the trajectory sets are used by algorithms to build the experimental models and, later, to evaluate them.
2.1. FUNDAMENTAL CONCEPTS

- **Training Set** is a random portion of trajectories used by algorithms for building the experimental models. After building the model, the trajectories and their locations are discarded and are not used anymore during the experiments. In my experiments, the training set is composed by 75% of the trajectories in the enriched GeoLife dataset.

- **Testing Set** is a random portion of trajectories used for evaluating the quality of the experimental models. In practice, these trajectories are the remaining ones of the entire dataset, which do not include any trajectory used in the building phase.

- **Learning approaches** define how the data (i.e., trajectories) were used to build the experimental models. More specifically, the experimental models were built considering two distinct approaches, namely a supervised and an unsupervised method.

  In the **supervised learning** approach, the training data is annotated with labels that represent the desired output, which in the context of the proposed task are the semantic labels. The goal of the supervised approach is to ensure the model learns how to link the input (i.e., symbols) to the output (i.e., semantic labels). For training the experimental model, the semantic labels are provided with other location information through the heuristic presented in Section 3.2.

  In comparison to the unsupervised approach, the main advantages of the supervised approach are the requirement of less data (e.g., less trajectories or visited locations) and, usually, the training of the model is faster. As expected there is a major disadvantage on the supervised approach, namely in the training of the model, that considers the expected labels for the visited locations on the training trajectories, and thus the visited location classification will take into account the assumptions associated to the training data (e.g., the existing assumptions on the heuristic used to annotate the visited locations). In addition, for previously unseen situations, a visited location can be classified with an arbitrary semantic label.

  In the **unsupervised learning** approach, the training data is not annotated and, iteratively, similar training data is grouped into a specified number of classes. In the unsupervised approach and accordingly to the model’s characteristics, the symbols that represent the visited locations of each trajectory are used to learn the parameters of the model. The learning process only ends when the used algorithm reaches an expected likelihood or a given number of iteration steps. In comparison to the supervised approach, the unsupervised method provides advantages such as the use of unlabeled data in the learning process (i.e., the labeled inductive bias is avoided) and the possibility to learn larger and more complex models. Concerning the disadvantages, we have the need of more data, or data collected in a manner oriented to the proposed task, for reaching the expected likelihood, as well as the
additional effort in the learning process.

2.1.1 Models for Sequential Data Classification

Sequential data classification is an extension of more traditional single instance classification problems, leveraging on sequential patterns so as to generate more accurate predictions. The problem concerns formally with receiving an input sequence \( x \), with \( n \) elements to classify, and producing a sequence \( y \) with semantic classes for each of the \( n \) elements in \( x \). Each of the elements in the input sequence can be described by several features, which will be used in the classification process. In this section, I present the most popular models that have been applied to solve sequential learning tasks. These sequence learning models include Sliding Window Classifiers, Hidden Markov Models or Conditional Random Fields (Nguyen & Guo, 2007).

2.1.1.1 Sliding Window Classifiers

The Sliding Window method converts sequential learning problems into classical single-instance supervised learning problems (Dietterich, 2002). It performs this simplification by partitioning the input sequence \( x \), with \( n \) elements, into several windows of width \( w \). For each of the windows, we consider an individual classification rule.

Consider that \( d \) is a numeric value corresponding to half of the window size \( w \) (i.e., \( d = \left\lceil \frac{w-1}{2} \right\rceil \)). Consider also that \( t \) indicates a position in the sequence of elements. A classifier \( h_w \) uses an input window \( (x_{t-d}, x_{t-d+1}, \ldots, x_t, \ldots, x_{t+d-1}, x_{t+d}) \) to predict the value \( y_t \). At the end, the input sequence is classified by concatenating each \( y_t \), resulting on the predicted sequence of labels. Note that the values outside each window are not considered for the prediction.

The possibility of combining the Sliding Window method with any classical supervised learning algorithm (e.g., state of the art ensemble methods such as Random Forests (Breiman, 2001) or CART (Breiman et al., 1984)) is an important advantage of this approach, and indeed Sliding Window Classifiers have been used in many different domains (Fawcett & Provost, 1997; Freitag, 1997; Qian & Sejnowski, 1988; Sejnowski & Rosenberg, 1987). However, Sliding Window Classifiers also have some limitations. For instance, they do not take into account the correlations existing between the different \( y_t \) values (e.g., for obtaining the value of \( y_t \), the previous \( y_{t-1} \) value is not used). The only relations between \( y_t \) values that are considered come from intersected windows with some common \( x_t \) elements.

By using a recurrent version of the Sliding Window method, one can improve classification accuracy by catching long-distance effects (Bakiri & Dietterich, 1997). The difference in this recurrent version relies on the most recent classified \( y_t \) values being used together with the other features
2.1. FUNDAMENTAL CONCEPTS

Figure 2.1: Execution of the Viterbi algorithm for the sequence AABC.

To predict the next \(y_{t+1}\) values. To predict the output value \(y_{t+1}\), not only we use input features derived from the observations in the window corresponding to \((x_{t-d}, x_{t-d+1}, ..., x_t, ..., x_{t+d-1}, x_{t+d})\), but also the previous classified \(y_t\) values, namely \((y_{t-d}, y_{t-d+1}, ..., y_{t-1})\).

There are several open-source implementations of Sliding Windows methods for addressing sequence classification problems, an example being the WEKA extensions\(^1\) for sequence classification problems.

2.1.1.2 Hidden Markov Models

The Hidden Markov Model (HMM) is a well-known generative probabilistic model for explaining the generation of sequences of observation symbols \(x\) associated with unobserved (i.e., sequences of hidden classes) states \(y\).

An HMM can be used to model a process, with a given duration \(T\), since it considers that each state \(y_t\) is directly influenced by the previous state \(y_{t-1}\), and linked to a corresponding symbol \(x_t\).

The HMM model also considers that each state can emit different symbols with distinct probabilities. In order to explain the generation of a sequence of symbols, the HMM takes into account three contributing factors. The first is the probability of the sequence starting at a particular state. The second is the probability of choosing a given state after another (i.e., transition probabilities). The last is the probability for the production of the different symbols in each state (i.e., emission probabilities).

Taking the notation of Dugad & Desai (1996), the model can be seen as a tuple \(\lambda = < A, B, \pi >\), inside the context of having \(N\) possible states and \(S\) different possible symbols. Respectively, the vector \(\pi = < \pi_{y_1}, ..., \pi_{y_N} >\) contains the probabilities of starting a sequence at a state \(y_n\). The matrix \(A = [a_{y_n, y_m}]\) contains the transition probabilities from a state \(y_n\) to a given state \(y_m\). The

\(^1\)http://web.engr.oregonstate.edu/~tgd/software/RSW/
matrix $B = [b_{yn,x}]$ contains the probabilities for the emission of symbol $x$, at a state $y_n$. Considering an instantiated Hidden Markov model as a tuple $\lambda = < A, B, \pi >$, the two problems that are more important to solve, in terms of sequence analysis, are (i) computing the probability of a given sequence $x$, that is computing $P(x|\lambda)$, and (ii) computing the most likely sequence of states $y$ to have generated an observed sequence of symbols $x$ (i.e., the classifications for symbols in $x$), that is, finding the state sequence $y = < y_1, y_2, ..., y_T >$ which maximizes $P(x,y|\lambda)$ for a given sequence of symbols $x = < x_1, x_2, ..., x_T >$.

Since the objective of the work reported in this dissertation is to provide contextual information about human behaviour, through the observation of trajectories, the focus in terms of hidden Markov modeling is on the second problem. Here, the Viterbi algorithm (Viterbi, 2006), which is a particular form of dynamic programming, addresses the problem of finding the most likely sequence of states for a given sequence of symbols. For each symbol $x_t$, the algorithm computes the probability of its emission for every possible state. The Viterbi algorithm starts by calculating the initial probability of the emission of symbol $x_1$ in all possible states. Then, for each state transition, it calculates again the emission of symbol $x_2$. This step is repeated for each symbol, until time $T$, where the given symbol sequence ends. Finally, with all possible paths covered, the algorithm finds the path that ends in the most probable state then goes from the end to the start collecting the most probable states, this way obtaining the most likely state sequence $y = < y_1, y_2, ..., y_T >$. The Viterbi algorithm can also solve the problem by considering costs, instead of probabilities, returning the sequence of states that minimizes the associated cost.

Many practical implementations follow this approach, by taking the logarithms of the probabilities as the costs.

Figure 2.1 presents an example for the execution of the Viterbi algorithm, using costs and considering an HMM with two states $y_a$ and $y_b$, and with a vocabulary of three symbols, namely $A$, $B$ and $C$. On the left, Figure 2.1 shows the cost values that are used in the example. The cost of the initial state, related to the vector $\pi$, is represented with small circles shaded in grey. The state transition costs, associated with the matrix $a_{yn,yn}$, are represented by arrows connecting the states. Finally, the costs associated with the emission of symbols (i.e., matrix $b_{yn,xs}$) are shown inside the state rectangles. On the right, Figure 2.1 illustrates the execution of the algorithm. In the first column, the figure shows the cost of emitting the first symbol $A$ on different states, that is $P(< x_1 = A >, y_1|\lambda)$. After that, and until the end of the given sequence of symbols, the algorithm calculates each state’s total accumulated cost for every possible transition path, at each step $t$. A state’s total accumulated cost is the sum of the symbol’s emission cost with the minimum value of the costs from each $t-1$ states (e.g., representing $P(< x_1 = A, x_2 = A, x_3 = B >, < y_1, y_2, y_3 > |\lambda)$, if in time $t = 3$). Each of these costs, from
2.1. FUNDAMENTAL CONCEPTS

the $t - 1$ previous states, is the state's total accumulated cost, plus the transition cost. In Figure 2.1, a tuple represents the state's total accumulated cost, at each step $t$. Within this tuple the values stand, respectively, for the minimum between the transitions from states $y_a$ and $y_b$, plus the emission of the symbol at that given state. Finally, at time $t = T$, the algorithm identifies the ending of minimum cost and executes a backtracking to discover the state sequence of minimum cost, for the given sequence of symbols.

The HMM approach has some limitations regarding the capture of correlations. The correlation between two separated states is one of these limitations, since states that are not connected cannot communicate directly with each other. This communication is made through other states that are in the path, represented by the black arrows in Figure 2.2. Although the first-order Markov model does not capture these kinds of long relations (e.g., in a first-order HMM, a state $y_t$ only depends on state $y_{t-1}$), higher-order models can minimize this problem. For instance, on a three-level model, we have that at a given time $t$, the probability of reaching a given state depends on the current state and on the two previous ones. This is represented in Figure 2.2 by the blue arrows between states.

The other limitation regards the generation of the symbols, since they depend only on the corresponding state $y_t$. This is a problem, since $P(x_t)$ may be related with the surrounding states of $y_t$. A possible representation for these missed relations is given by the red arrows in Figure 2.2. Theoretically, in HMMs, one can use a distribution like $P(x_t | y_{t-1}, y_t, y_{t+1})$, so that $P(x_t)$ can be influenced by the surrounding values in $y$. However, the principles behind this representation are not as clear as in traditional HMMs.

To apply HMMs to our task, we consider the information available in each trajectory's location. Therefore, a symbol of the HMM is a representation of a trajectory's spatio-temporal location and, in this way, a temporal sequence of symbols stands for a trajectory. For building each location's representation (i.e., the corresponding symbol), we use both the geospatial and the temporal information.

For detailed information about the usual algorithms for solving problems with hidden Markov
models, please refer to the tutorial by Rabiner (1990). As with Sliding Window methods, there
are some open-source implementations of the algorithms associated to Hidden Markov Models,
such as HMMWeka\(^1\) and JaHMM\(^2\).

### 2.1.1.3 Conditional Random Fields

Conditional Random Fields (CRFs) are discriminative undirected graphical models, where the
labels \(y\) for a given observation sequence \(x\) can be selected by maximizing the conditional
probability \(P(y|x)\) (Lafferty \textit{et al.}, 2001).

The main idea behind the conditional model is to specify the probabilities of possible label se-
quences, given an observation sequence, contrasting with a joint distribution over both obser-
vation and label sequences, as in HMMs. The conditional nature of the model means that no
effort is wasted on modelling the observations, and it is easier to represent the data without
making independence assumptions (i.e., arbitrary attributes of the observation sequence may
be captured without taking into consideration concerns about how these attributes are related
(Wallachi, 2004)). Although the general conditional idea can be adjustable to specific problems,
for sequence labelling one normally uses linear-chain CRFs. The graphical representation of this
model is a chain structure composed by one vertex for each random variable \(y_i\), and edges show-
ing the dependency between two random variables, namely \(y_i\) and \(y_{i-1}\). These random variables
\(y_i\) will be inferred through the variables of the observed and fixed input sequence \(x\). Note that
each random variable \(y_i\) must respect the Markov property. Intuitively, this property means that
the neighbours of \(y_i\) contain all the necessary information to predict its value.

The conditional probability distribution of a label sequence \(y\), given an observation sequence \(x\),
is written as:

\[
P(y|x, \lambda) = \frac{1}{Z(x)} \exp \left( \sum_{j \in J} \left( \lambda_j \sum_{i=1}^{\left| x \right|} f_j(y_{i-1}, y_i, x, i) \right) \right)
\]

In the equation, \(Z(x)\) is a normalization factor that ensures the distribution \(P(y|x, \lambda)\) sums up to
1. This model considers a given and fixed set of features \(J\), and each feature \(j\) expresses some
characteristic (i.e., an attribute or property) of the observation sequence (e.g., if the location was
visited in the evening) through real-values. So, each \(f_j(y_{i-1}, y_i, x, i)\) is a feature function, either a
state function or a transition function from \(y_{i-1}\) to \(y_i\), of the label at position \(i\) and that may use the
entire observation sequence. The parameter \(\lambda\) is a vector estimated from the training data. This
can be done via two different types of methods, namely iterative scaling (Berger, 1997; Darroch

---

\(^1\)http://www.doc.gold.ac.uk/~mas02mg/software/hmmweka/index.html
\(^2\)http://www.run.montefiore.ulg.ac.be/~francois/software/jahmm/
2.2. RELATED WORK

& Ratcliff, 1972; Pietra et al., 1995) or gradient-based methods (Sha & Pereira, 2003; Wallach, 2002).

The advantages of CRFs are many, considering other sequence models. One of these advantages is the fact that CRFs are better suited to include rich and overlapping features. In a feature function concerning an arbitrary random variable, one can include information about other observation elements that are not directly connected with the random variable. Another advantage is related with not modeling the input sequence $x$, which often contains many highly dependent features that are difficult to model.

CRFs typically outperform HMMs on sequence labelling tasks (Lafferty et al., 2001; Pinto et al., 2003; Sha & Pereira, 2003). There are several available open-source implementations of CRF models for addressing sequence classification problems. These include CRF++, MALLET, CRFSuite and FACTORIE.

2.2 Related Work

Previous works have addressed distinct issues involving trajectory data. The book by Zheng & Zhou (2011) has presented an extensive and detailed survey.

Parent et al. (2013) explained the basic concepts involved in trajectory analysis, and they also discussed several techniques for the enrichment of trajectories with semantic meaning, and for the extraction of behaviour knowledge from trajectories. Trajectory enrichment concerns with the addition of contextual knowledge to the raw data, from a contextual data repository (e.g., from geo databases). The contextual data links happen either during or before the trajectory’s analysis. The survey of Parent et al. has a particular focus on data mining techniques, although there are mentions to other distinct approaches considering the manner through which data are modeled, such as state-space models or hybrid methods, where the last ones typically combine template matching techniques with other types of strategies.

2.2.1 Data Mining Techniques

Data mining techniques focus on discovering information patterns in data sets. The overall goal of a data mining process is to extract information from a data set and, then, transform it for further

---

1http://crfpp.sourceforge.net/
2http://mallet.cs.umass.edu/
3http://www.chokkan.org/software/crfsuite
4http://www.factorie.cc
use on a specific work or task. The available methods have a wide range of application, since they can work in both low and high level contexts. Among the tasks performed by data mining methods are the analysis over raw data, the data pre-processing, or visualization of the results from distinct information perspectives.

For the type of tasks related to my thesis proposal, the data mining techniques normally consider each trajectory to be an ordered set, since sequence analysis methods or modified versions of the Apriori algorithm are typically used (Monreale et al., 2009; Morzy, 2007). Frequent patterns or association rules are identified by first arranging the trajectories into small location sets, and then the analysis methods iteratively evaluate the rules with a pre-determined confidence support. At the end, the resulting patterns and association rules are used to classify the locations within the trajectories. Data mining techniques tend to perform less accurately since they only rely upon the available data, specially in situations where the dataset does not cover most of all possible cases, or when there exists a high possibility of uncertainty. Furthermore, data mining methods do not usually consider the spatio-temporal distances between the timestamped locations within the trajectories.

Morzy (2007) presented a data mining approach to predict the future location of a moving object. The first step of his approach is extracting association rules from the moving object dataset. Then, for a previously unknown trajectory, the proposed approach uses matching functions to choose the best association rule that fits to this trajectory, and afterwards makes the prediction based on it. The author reports an accuracy of 80% for the best configuration of the system.

2.2.2 Sequence Classification Approaches

State-space models try to identify the differences between spatial sequences through sequence classification models, such as generative Hidden Markov Models (HMMs) (Rabiner, 1990), discriminative Conditional Random Fields (CRFs) (Sutton & McCallum, 2012; Vail et al., 2007), or more complex versions of these two well-known approaches (Dietterich, 2002; Nguyen & Guo, 2007). In opposition to data mining approaches, sequence models can deal with uncertainty, since they are based on probabilities. Both HMMs and CRFs are distinct but valid approaches for the proposed work of trajectory classification.

Asahara et al. (2011) proposed a state-space method for predicting pedestrian movement on the basis of a mixed Markov-chain Model (MMM). The proposed method is distinct from other previous approaches for movement prediction because it takes into account the pedestrian’s personality (i.e., a hidden state influencing the observations) and his previous status. The effectiveness of the proposed method is demonstrated in a shopping mall experiment. In the conducted
experiment, the MMM method surpasses more conventional methods, namely the Markov-chain Model (MM) and the Hidden-Markov Model (HMM).

In the case of the MM, pedestrians choose their next positions only based on their current positions. On the other hand, when considering the HMM, the unobservable states can be seen as either the personality or the thoughts of the pedestrian, which are related to the previous thoughts (i.e., other unobservable states). The observed symbols are viewed as the measured positions, which are changed only by the corresponding thoughts (i.e., hidden unobservable states). The proposed MMM method is an improved MM which instead of having an unobservable parameter for each position, corresponding to only a pedestrian as in the HMM, has a single parameter representing a pedestrian group.

The hidden parameter in the MMM is estimated from the available transition histories of pedestrians, as seen in Figure 2.3, which are reflected in a latent vector $Z_k$ and where $K$ represents the number of different pedestrian categories and their corresponding model versions (i.e., one model version by each pedestrian category). The categorical parameter is discovered through an expectation-maximization (EM) algorithm, which is composed by the E and M steps (Dellaert, 2002; Schönh, 2009). As shown in the Equation 2.2, the E-step is used for calculating the expectation values of $Z_k$ based on $\gamma_k(x)$.

$$\gamma_k(x) = \frac{\pi_k \prod_{u,v} w_{k,uv}^{x_u,x_v}}{\sum_{k'} \pi_{k'} \prod_{u,v} w_{k',uv}^{x_u,x_v}}$$

Equations 2.2, 2.3 and 2.4 are used for updating the model parameters, namely the mixing coefficient $\pi_k$, and the transition probabilities $w_{k,uv}$, from the position $u$ to position $v$, considering $k$ as the model version.

$$\pi_k = \frac{1}{N} \sum_{|x|} \gamma_k(x)$$

$$w_{k,uv} = \frac{\sum_{|x|} x_u x_v \gamma_k(x)}{\sum_{u,v} x_u x_v \gamma_k(x)}$$

Equations 2.2, 2.3 and 2.4 determine the parameters and the latent vector $Z_k$ that maximize the
likelihood, building the MMM model. To predict the next action of the pedestrian with maximum probability, one uses Equation \ref{eq:prediction}. The maximum likelihood is the sum of the likelihoods given by all $K$ models versions (i.e., representing all pedestrian categories) weighted with $\gamma_k(x)$.

$$\arg \max P(x_t|x_{t-1}, w_{k,uv}, \pi_k) = \arg \max \sum_k \gamma_k(x_{t-1}) \prod_{u,v} w_{k,u,v}^{x_u x_v} \tag{2.5}$$

The evaluation of the proposed MMM method was done through two distinct datasets, namely one simulating the behavior of pedestrians and another collected in the field. The simulation dataset only reflected the layout of the building, since the simulated pedestrians do not have preferences. Only in the field dataset, the preferences were revealed through movement bias. In both datasets, the MMM model overcomes the MM and the HMM models, since they do not consider either layout constraints or the common preferences of the pedestrians.

Results show that the MMM method outperforms all others. More precisely, in the simulation dataset, the prediction rate achieved by the MMM was 74.1%, contrasting with the 16.9% and 4.2% of the MM and HMM approaches, respectively. For the dataset collected in the field, MMM achieved 64.0%, contrasting with the 45.6% and 2.41% of both the MM and HMM models.

In the subsequent study, Asahara \textit{et al.} (2012) proposed a state-space modeling method for predicting pedestrian movement based on the HMM model, namely the mixed autoregressive HMM (MAR-HMM). Although the HMM is a flexible approach, the authors identified a problem related to the predicted positions, concerning the fact that the individual positions do not depend on the previous ones. Differently from what the HMM model does for the unobservable states (i.e., the Markov chain), the positions are not dependent on the previous ones due to the nonexistence of direct dependencies between the generated symbols. Before introducing the MAR-HMM, the authors presented the autoregressive HMM (AR-HMM) model to face the exposed limitation. In opposition to the traditional HMM, the AR-HMM model adds a chain between the symbols (i.e., positions) at the cost of considering a larger number of parameters. The MAR-HMM is a special case of the AR-HMM, where the authors dissociate pedestrian stable properties (e.g., age and gender) from the unobservable states of the AR-HMM model, thus creating a second level of states that resembles the mixed Markov-model (MMM) from their previous work (Asahara \textit{et al.}, 2011). By grouping the unobservable states that share the same stable properties, the top-level states constrain and reduce the number of unobservable states and the transitions between them. Therefore, the parameter estimation is simpler on the MAR-HMM model.

The authors compared the proposed method to others, namely the MMM and the AR-HMM. In that comparison, the proposed method outperforms the others by achieving 56.8% accuracy,
2.2. RELATED WORK

Liao et al. (2007) proposed a CRF-based approach to extract personal activities and significant places from traces of GPS data. More specifically, a hierarchical CRF was used to infer the everyday activities (e.g., working or travel) and to label significant places (e.g., workplace or friend’s house) of the user. The proposed approach overcomes two limitations identified in previous works, namely the restrictions in traditional activity models where some relevant information was not considered (e.g., different types of places or motion between places) and the inaccurate place detection due to time thresholds.

The hierarchical CRF is executed in three phases, namely the conversion of GPS measurements to street map points, inferring activities and types of significant places, and place detection and labeling. In the initial phase, the GPS points are segmented to represent activities that occurred in certain areas. If a discretized street map is available, the GPS points are associated with the map patches (i.e., 10 meters section of a street) by a CRF model. Otherwise, the segments are defined by a pre-determined distance (e.g., in the context of this work, a segment was formed by 10 meters of consecutive GPS points). The discretized street map usage has the purpose to refine the GPS point segmentation, improving the inference of activities. The CRF model structure is shown in Figure 2.4, where GPS points \( g \) are associated with map patch \( s \) (i.e., corresponding to the hidden states) within a given duration \( T \). The value for each \( s_i \) ranges over the street patches in the map that are within a certain distance of the GPS point \( g_t \). In order to obtain the conditional probability of the CRF model that refines the GPS segmentation using the map patches, the authors defined three sub-graphs (i.e., cliques) and their respective feature functions to characterize the spatio-temporal relations between them. All the cliques are represented by greyish areas in Figure 2.5. The GPS noise and map uncertainty are considered in the dark grey clique. The following feature function, considering the previous spatial information, measures the squared distance between a GPS reading and the center of a street patch, where \( \sigma \) controls the scale of the distance.

![Figure 2.4: The concept hierarchy for location-based activity recognition (adapted from Liao et al. (2007)).](image-url)
CHAPTER 2. CONCEPTS AND RELATED WORK

Figure 2.5: CRF for associating GPS measurements to street patches (adapted from Liao et al. (2007)). The different types of cliques are shown in shaded areas.

\[ f_{\text{meas}}(g_t, s_t) = \frac{\|g_t - s_t\|^2}{\sigma^2} \] (2.6)

Temporal consistency is ensured by light grey cliques with four nodes. The next feature function compares the spatial relationship between consecutive GPS points and the spatial relationship between their associated patches. The more similar the spatial relationship, the more consistent is the association.

\[ f_{\text{cons}}(g_t, g_{t+1}, s_t, s_{t+1}) = \frac{\|((g_{t+1} - g_t) - (s_{t+1} - s_t))\|^2}{\sigma^2} \] (2.7)

The smoothness between street patches is represented by medium grey cliques. The intuition of the considered cliques is to prefer traces that do not change frequently between streets. The following binary feature function is used to test whether consecutive patches are on the same street, in neighboring streets, or in the same direction.

\[ f_{\text{smooth}}(s_t, s_{t+1}) = \delta(s_t.\text{street}, s_{t+1}.\text{street}) \cdot \delta(s_t.\text{direction}, s_{t+1}.\text{direction}) \] (2.8)

The function \( \delta(u, v) \) corresponds to 1 if \( u = v \) and 0 otherwise. Thus, the CRF models consider the presented feature functions to infer the conditional distribution between street patches and GPS traces:

\[ p(s|g) = \frac{1}{Z(x)} \exp \left\{ \sum_{t=1}^{T} w_m \cdot f_{\text{meas}}(g_t, s_t) + \sum_{t=1}^{T-1} (w_c \cdot f_{\text{cons}}(g_t, g_{t+1}, s_t, s_{t+1}) + w_s \cdot f_{\text{smooth}}(s_t, s_{t+1})) \right\} \] (2.9)

In the equation, \( Z(x) \) is a normalization function. In order to build a generic model that handles different persons and locations, the feature function weights do not depend on the time index. The authors used the Parameter Sharing technique to learn the weights for each type of feature function, thus avoiding learning for each location the specific best classifying weights.
2.2. RELATED WORK

After the GPS point segmentation, a CRF is built for inferring the corresponding activity for each spatial segment. The activities are observed as being from two different types, namely navigation and significant ones. Considering that the navigation activities imply the movement between locations (e.g., walking or driving), the significant ones occur when a user stays at a location (e.g., work, visit, drop off, or pickup).

To determine the activity for each GPS segment, the CRF model considers four indicator features. The indicator features include temporal information (e.g., day of the week), average speed for discriminating the distinct transportation modes, location information extracted from geographic databases (i.e., indicating if the segment is on a bus route, whether it is near a local business or bus stop) and the compatibility with the activities of nearby nodes. As presented in Figure 2.4, the aforementioned CRF composes the lower structure of the proposed hierarchical CRF.

Finally, the activities and the GPS segments are iteratively used to detect the significant places, which are locations that represent a significant role in the activities of the user (e.g., home or work place). Considering that initially the significant places are not known, annotations are generated through an algorithm that receives an activity sequence and determines to whether or not individual activities belong to a significant place. In addition, and since a place can be visited multiple times within an activity sequence, spatial clustering is also performed by the algorithm to remove duplicate places. The received activity sequence is the most likely configuration for the user’s GPS segments and it is obtained by applying the maximum posterior inference (MAP) approach over the lower CRF. Then, a complete version of the hierarchical model is built with the generated places, the activities and their GPS segments.

In the complete CRF, the significant places are inferred assuming that the activity which occurred in a given place strongly indicates its type, and assuming a limited number of homes and workplaces. In practice, the first assumption is executed by obtaining an indicator feature for each combination of the type of place, type of activity and frequency category. For calculating the indicator features, a clique is generated for each place that contains all activity nodes in its vicinity. Then, the activities occurring in a given place are counted and divided into weekly frequency categories (e.g., $2 \leq \text{count} \leq 3$). Regarding the second assumption, the features are simply the counts of different homes and workplaces. These features improve the extraction by exponentially decreasing the likelihood as the counts increase.

Taking into account that a different MAP activity sequence can be generated from the complete CRF, a MAP estimation is performed on the new CRF. The final phase is repeated until the MAP activity sequence does not change. For each new iteration, the process starts by generating the significant places, but now considering the MAP activity sequence of the complete CRF. Ultimately, the algorithm returns the MAP activity sequence and the corresponding set of places and
2.2.3 Hybrid Approaches

Hybrid approaches normally combine different strategies with template matching techniques, which are based on performing a similarity search on sequential data. For finding the similarities in sequences or time-series data, specific metrics evaluate the extracted features against previously identified patterns or templates. The used metrics are mainly adaptations of algorithms used in traditional string matching problems, such as edit distance or longest common subsequence, although dynamic time warping or other heuristic algorithms can also be used (Hoda & Mokhtar, 2009; Riedel et al., 2008). Template matching techniques normally have a long training time and a low sensibility for spatial variation, due to the inclusion of partial searches using subsequences of a given sequence.

Ying et al. (2011) proposed an approach for predicting the next location of a given user, based on the geographic and semantic features of the trajectories. The proposed model performed the prediction task using a strategy that combines the features of a given trajectory with clusters and pattern analysis. Figure 2.6 illustrates how the framework of the proposed method performed the their MAP types.

The experimental results shown that the proposed approach outperforms others. Apart from learning a person's significant places, the proposed model can infer low-level activities (e.g., walking or picking a bus). The hierarchical CRF achieved accuracies of 85% in low-level activity inference, and 90% accuracy in detecting and labeling significant places.
2.2. RELATED WORK

Figure 2.7: An example of cell semantic trajectories (adapted from Ying et al. (2011)).

prediction task. More specifically, the SemanPredict framework divided the prediction process into two modules, namely the Offline Training Module and the Online Prediction Module.

The Offline Training Module has the purpose of discovering trajectory patterns by observing both geographic and semantic behavioral information of the users. For discovering the trajectory patterns, this module uses a three-phase pipeline, orderly composed by Data Preprocessing, Semantic Mining and Geographic Mining steps. Initially, in the Data Preprocessing step, the trajectories are transformed into stay location sequences, which are locations where a user stopped for some time. In the context of this work, stay locations are obtained using two different methods since the trajectories can be represented by two distinct types of data, namely GPS or cell trajectories. When considering GPS trajectories, a density-based clustering algorithm is used to discover the stay locations within the points of the trajectories (Zheng & Zhou, 2011). When handling cell trajectories to obtain stay locations, the cells are filtered with time and crowd thresholds. The time is given by the difference between the user arrival and departure timestamps of the cell, while the crowd threshold defines the required number of users that passed on the cell.

The Semantic Mining step extracts the semantic trajectory patterns from the stay location sequences of the user, and groups similar users into clusters. The approach from Ying et al. (2010) is followed to determine the semantic trajectory patterns, where the Maximal Semantic Trajectory Pattern Similarity algorithm is employed to measure the semantic similarity between trajectories. The semantic labels that are attached to the stay locations come from a geographic semantic information database (GSID). The used GSID contains information collected from Google Maps, such as landmarks, their semantic scopes and the associated semantic labels. As Figure 2.7 illustrates, a stay location overlapping one or several landmarks is associated to their
semantic labels. Otherwise, the *Unknown* semantic label is assigned to the stay location. After transforming each stay location sequence into a semantic trajectory and building a semantic trajectory dataset, the Prefix-Span mining algorithm (Pei et al., 2004) is used to identify the semantic trajectory patterns (STP). An example of a STP, where a stay location overlaps two landmarks (i.e., *Bank* and *Park*) at the same time, could be \(<\text{Unknown}, \text{School}, (\text{Bank, Park}), \text{Hospital}>\). Although the semantic label of the next location can be predicted by matching the recent moves of a mobile user with the STPs, there is a gradual loss of efficiency with the increase in size of the pattern (i.e., more subsequences will be considered for the prediction when the pattern is longer). A semantic trajectory pattern tree (STP-Tree) method, based on a prefix tree structure, was used to bypass the aforementioned limitation and represent the indexed collection of STPs. In the STP-Tree, a STP is considered as a sequence of semantic labels and each one of them is associated with a support value. The support value stands for the number of ordered sets that can be extracted from the pattern set, divided by the total number of trajectory patterns in the dataset. Since each path on a STP-Tree indicates a decision rule, the longest prefix is searched in the tree when given a STP. When the searched path is only a prefix of the given STP, the remaining elements are appended to the STP-tree and the support values for the STP elements are updated. If the initial element of the given STP is not included in the STP-Tree, the STP is added to the root of the tree. Considering the STPs \(<\text{School, Park, Hospital}>> and \(<\text{School, (Bank, Park), Restaurant}>>\), one can extract sets like \(<\text{Park, Hospital}>, <\text{Hospital}>, or <\text{Park}>>\), with the respective support values associated of 1/2, 1/2 and 2/2.

When predicting the next location of a given user, the authors explored the semantic behavior of similar users, instead of only using the past behavior of the user. The similar behavior between users is captured by a heuristic which states that the trajectories are more similar when they have more parts in common. The Longest Common Sequence (LCS) algorithm is applied to the STPs of different users to identify the common parts. Finally, the weighted average of all possible common parts between the STPs of two users determines if they should be part of the same cluster.

Although locations cannot be determined through semantic labels, the clusters based on STPs helped to aggregate the stay location sequences of similar users. Hence, the authors avoid the use of the frequent behaviors of all users and, possibly, an imbalanced data problem. In the following Geographic Mining step, and constrained to the clusters of similar users, the Prefix-Span mining algorithm is used again to find the sequence location patterns and to build the corresponding prefix tree, called the Stay Location Pattern tree (SLP-tree). Later, the prediction of the next location of the user occurs in the On-Line Prediction Module, where the STP-tree
and a cluster-based SLP-tree are used to generate candidate paths. The candidate paths are evaluated by a weighted average of two scores, namely a geographic score and a semantic score.

\[
\text{score} = \beta \cdot \text{geographic score} + (1 - \beta) \cdot \text{semantic score}, \quad \text{where } 0 < \beta \leq 1 \tag{2.10}
\]

In order to evaluate candidate paths, the current trajectory of the user is transformed into a stay location sequence. Considering that using all possible subsequences of the given trajectory would consume much time, the authors proposed a partial matching strategy. The partial matching strategy implied three conditions, namely (1) outdated moves may potentially deteriorate the precision of predictions, (2) more recent moves potentially have more important impacts on predictions, and (3) matching paths with a higher support and a higher length may provide a greater confidence for predictions. Having the stay location sequence and its subsequences, the cluster-based SLP-tree is searched to access the geographic score of the candidate paths. Figure 2.8 illustrates the depth-first search algorithm applied to the SLP-tree and used to compute the geographic score. Equation 2.11 defines how the geographic score is calculated for a given stay location sequence \( S \) when matching a path \( P \) in the SLP-tree.

\[
\text{geographic score}(P, S) = \sum_{i=1}^{\vert P \vert} \sum_{j=k}^{\vert S \vert} \alpha^{\vert S \vert-j} \times m\text{score}(P_i, S_j) \tag{2.11}
\]

\[
m\text{score}(P_i, S_j) = \begin{cases} P_i \cdot \text{support} & \text{if } S_j \text{ is matching to } P_i \\ 0 & \text{otherwise} \end{cases} \tag{2.12}
\]

The parameter \( \alpha \) represents the over time decrease of influence for each matched stay location of the evaluated pattern, thus applying the conditions (1) and (2) of the partial matching strategy. Following the example presented in Figure 2.8 and according to both the depth search algorithm...
and the formula on Equation 2.11, the stay location sequence of the user generates three candidate paths, as shown in Table 2.1. Table 2.1 presents \(<StayLocation_0, StayLocation_1>\) as the longest sequence pattern founded when considering the parameter \(\alpha = 0.8\). After finding the candidate paths and calculating the geographic score, the candidate paths are converted to semantic paths in order to calculate the semantic score. The semantic score of candidate paths is calculated equivalently to the geographic score, since Equation 2.11 is used again and the depth search algorithm is applied to the individual STP-tree of the user. Finally, the final score of each candidate path is calculated by setting the \(\beta\) parameter and knowing both the geographic and the semantic scores. Then, the child location of the candidate path with the highest score is considered the predicted location. In case the candidate path does not have a child location, the predicted location goes to the child location of the candidate path with the second highest score, and so on.

The proposed method was evaluated through experiments with the reality mining dataset from the MIT (Nathan Eagle & Lazer, 2009), which is mostly a cell trajectory dataset containing 106 users and over 500,000 hours of continuous daily activities. The test phase opposes the SemanPredict method to other non-clustering approaches, namely a Geographic Only (i.e., not containing semantic information) and a Full-Matching (i.e., instead of using a partial match strategy, it considers all possible subsequences of the sequence) baseline. Results show that the SemanPredict method outperforms all others, except on precision when comparing to the Full-Matching approach.

<table>
<thead>
<tr>
<th>Candidate paths</th>
<th>geographic score</th>
</tr>
</thead>
<tbody>
<tr>
<td>(StayLocation_3)</td>
<td>0</td>
</tr>
<tr>
<td>(StayLocation_0) → (StayLocation_1)</td>
<td>0.8 × 0.667 + 0.667 = 1.2</td>
</tr>
<tr>
<td>(StayLocation_1)</td>
<td>1.0</td>
</tr>
</tbody>
</table>

Table 2.1: Candidate path sets and the respective geographic scores (adapted from Ying et al. (2011)).

Participants in the Nokia Mobile Data Challenge (MDC) event presented an approach to address the problem of classifying places visited within trajectories (Laurila et al., 2012). In this event, and for the visited places classification task, the winning method by Y. Zhu & Yang (2012) showed the vital importance of feature engineering, before using a well-known model. The authors proposed the use of a technique based on conditional features, which involved the evaluation and the generation of two types of features, namely the unconditional and the conditional features. Each unconditional feature is a single value (e.g., WiFi strength). As for the conditional features, the correlation and dependency between variables were explored where one or more temporal
2.3. SUMMARY

features conditioned the calculation of a non-temporal one (e.g., WiFi strength conditioned by WeekDay/Weekend). The technique selected a small set of features thought to be more important for obtaining a correct classification with the model by analyzing the relationship between them. An important conclusion to retain was that the time dependent features being very useful for the place category classification. The models that had better results were based on Gradient Boosted Trees and L1-regularied Logistic Regression, achieving respectively an accuracy of $75.1\%$ and $74.6\%$.

2.3 Summary

Section 2 described the main concepts involved in my thesis proposal, presented the models that can address the classification of trajectories and detailed the most relevant related work.

Despite of the increasing popularity of geopositioning information, the most previous work has essentially focused on spatio-temporal features of trajectories, instead of trying to extract semantic meaning from the trajectories, which is associated with human activities and routines. In addition, the analysis of the related work showed that there are many different challenges associated with the classification or prediction tasks.
Chapter 3

Semantic Classification of Locations within Trajectories

This chapter presents different methods for semantically classifying locations visited in the context of a trajectory. I start by giving a formal overview about this problem, focusing on the input data and on the information resulting from a classification approach. Then, I explore different methods for executing the classification task, while also analyzing their distinct features.

3.1 Overview

A trajectory is an ordered set of visited locations collected by a GPS receiver. Formally, a trajectory is represented as a sequence \( t = < l_0, ..., l_n > \), in which \( n \) is the number of locations. Note that the number of locations may vary from one trajectory to another. Within a trajectory, each of the locations is characterized by a tuple \( l_i = < \text{latitude}, \text{longitude}, \text{timestamp} > \). The first two fields are the coordinates of the position captured by the GPS receiver, and the last one registers the temporal instant when the visit happened. An entire trajectory dataset can be represented as \( T = < t_0, ..., t_m > \), where \( m \) is the number of available trajectories.

Using a trajectory dataset as described above, we are only able to identify spatial and temporal patterns. Since the objective of our work is to analyze human behavior, we need to add some information about the categories of the individual locations within these raw data. Nowadays, location-based social networks disseminate this missing information. In a location-based social network (LBSN), people share information about visited locations and, normally, this includes specific contextual information. Some of the existing LBSNs suggest tags to users, in order to
represent these contextual elements. A tag usually helps to identify immediately the activities that can be performed in a specific location. For instance a tag can be a category (e.g., Food), which may indicate the presence of a restaurant in a specific location. To include the behavioral information in a trajectory dataset, we use the location’s coordinates (i.e., latitude and longitude) to look for contextual information in a specific LBSN, namely Foursquare. Then, we add a new field to each location in the trajectory dataset, containing the search result for the location’s raw information. Each location in the original trajectory dataset is thus afterwards represented as a tuple $l_i = < \text{latitude}, \text{longitude}, \text{timestamp}, \text{cat} >$, in which $\text{cat}$ refers to the semantic category of each location.

By associating semantic categories to existing trajectory data, and latter through automated analysis, we can perhaps address questions such as (1) what are the categories that are typically visited in sequence within the trajectories, or (2) which categories are most popular, at particular geographic regions or at particular temporal intervals?

### 3.2 Heuristic Trajectory Classification

For performing trajectory classification, we can use the coordinates of each location to search for behavioral information in a LBSN. Since geopositioning devices collect the locations associated with a timestamp, we can also strengthen an heuristic classification approach with this information. Believing there are activities that are more likely to happen at a particular time of the day, I propose to use time intervals to filter the categories representing the activities. Our common sense determines these time intervals, believing that they can apply to different cultures. In this way, we built a classification heuristic that will try to take full advantage of the available information, for each specific point in a trajectory.

The main elements of LBSNs are places, where users do and share activities. Thus, we are able to associate the geographic characteristics of these places to behavioral information. Normally, LBSNs provide users two ways of describing their activities in a specific place, namely through comments, and by selecting existing tags. We use the tags as a semantic meaning (i.e., categories) for the locations.

We choose the Foursquare\(^1\) LBSN for collecting behavioral data because of its popularity, verified by both the number of registered users and by the number of indexed places, with basis on the information provided by Foursquare. We considered a set of nine possible categories corresponding to $\text{cat} \in \{ \text{Shop and Service, Nightlife Spot, College and University, Food, Outdoors} \}$

\(^{1}https://developer.foursquare.com/\)
3.2. HEURISTIC TRAJECTORY CLASSIFICATION

and Recreation, Professional and Other Places, Arts and Entertainment, Residence, Travel and Transport]. Due to the number of existing locations and due to limitations in FourSquare’s API, we developed a cache mechanism to be used in the data collection process, in order to avoid similar queries to FourSquare’s API. We also considered adding the same tags for locations within a 70 meters radius of a given point, without performing an additional search.

Considering both the location's coordinates and the timestamp, the dataset is enriched with the previously mentioned set of categories from FourSquare. The algorithm described next supported the choice of one of these categories to label each GPS location within a trajectory. For each of the trajectories in a given dataset, the following steps are applied:

1. Use the Foursquare API to collect the set \( S \) of all local businesses that are located within a radius of 150 meters from the geospatial coordinates associated to the trajectory point;

2. Return UNKNOWN if \( S = \emptyset \), and consequently discard the current location. We can then re-start the process for the remaining part of the current trajectory;

3. Return the category associated to the closest local business in \( S \), if this category \( \neq \) UNKNOWN;

4. Return Nightlife Spot if the name for the closest local business in \( S \) contains the corresponding keywords presented in Table 3.2, and if the timestamp associated to the trajectory point is within a period of the day between 22:00PM and 05:00AM;

5. Return Arts and Entertainment if the name for the closest local business in \( S \) contains the corresponding keywords presented in Table 3.2.

6. Return Shop and Service if the name for the closest local business in \( S \) contains the corresponding keywords that are presented in Table 3.2.

7. Return Food if the name for the closest local business in \( S \) contains the corresponding keywords that are presented in Table 3.2.

8. Return the category associated to the second closest business in \( S \), if this category exists and if it is \( \neq \) UNKNOWN;

9. Return the most visited category, in the set of local businesses \( S \), if this category \( \neq \) UNKNOWN;

10. Return the category associated to the most visited local business in \( S \), if this category \( \neq \) UNKNOWN;
3. SEMANTIC CLASSIFICATION OF LOCATIONS WITHIN TRAJECTORIES

<table>
<thead>
<tr>
<th>Category</th>
<th>Keywords</th>
</tr>
</thead>
<tbody>
<tr>
<td>Nightlife Spot</td>
<td>bar, pub, disco, or club.</td>
</tr>
<tr>
<td>Arts and Entertainment</td>
<td>cinema, movie, theater, or hall.</td>
</tr>
<tr>
<td>Shop and Service</td>
<td>store, shop, or mart.</td>
</tr>
<tr>
<td>Food</td>
<td>restaurant, bistro, café, or teahouse.</td>
</tr>
</tbody>
</table>

Table 3.2: Category keywords.

11. Return \textit{Nightlife Spot} if this category exists in the set of local businesses $S$, and if the timestamp associated to the trajectory point is within a period of the day between 22:00PM and 05:00AM;

12. Return the category associated to the second most visited local business in $S$, if this category exists and if it is $\neq$ UNKNOWN;

13. Return the second most visited category, in the set of local businesses $S$, if this category exists and if it is $\neq$ UNKNOWN;

14. Return UNKNOWN, ignoring this location and transforming the annotated part of the current trajectory (i.e., the annotated locations until the current location, which is to be discarded) into a new and individual trajectory, if it contains at least two already annotated locations. Then, we re-start the algorithm for the remaining part of the current trajectory (i.e., starting with the next location that is not annotated), which we consider to be also a new and individual trajectory, if it contains at least two already annotated locations.

From the previous algorithm, it is important to notice that we will split a trajectory into two separate ones when there exists one location that was not possible to annotate with a corresponding category. We considered this approach in order to use most of the available data.

3.3 Human Trajectory Classification by Using Hidden Markov Models

In this section, we present the proposed methods based on Hidden Markov Modeling. Hidden Markov Models (HMMs) were already detailed in Section 2.1.1.2.

When presenting the challenges involved in this task, the considered notation is the one introduced on Section 2.1.1.2. Under a context of $N$ possible states and $S$ possible symbols, the model is seen as tuple $\lambda = \langle A, B, \pi \rangle$. This model contains the probabilities of starting a sequence in a specific state $y_n$ as the vector $\pi$. The $A$ parameter represents a matrix containing...
the transition probabilities from a state \( y_n \) to a given state \( y_m \). Lastly, the matrix \( B \) contains the probabilities for the emission of symbol \( x_s \) at a state \( y_n \).

Since the objective of the work reported in my MSc dissertation is to provide contextual information about human behavior, through the observation of trajectories, the focus in terms of hidden Markov modeling is on finding the most likely sequence of states for a given sequence of symbols. As presented and exemplified/illustrated in Section 2.1.1.2, this classification task is addressed using the Viterbi algorithm (Viterbi, 2006).

To apply HMMs to our task, we consider the information available in each trajectory’s location. Therefore, a symbol of the HMM is a representation of a trajectory’s spatio-temporal location and a temporal sequence of symbols stands for a trajectory. For building each location’s representation (i.e., the corresponding symbol), we use both geospatial and temporal information.

See Section 2.1.1.2 for further details about HMMs, related algorithms, and a discussion on the limitations associated to this model. For examples of its practical application, start in Section 2.2.2.

### 3.3.1 Handling Spatio-Temporal Positions Through a Triangular Mesh

Considering the continuous spatio-temporal data in the GeoLife dataset, and considering that HMMs typically use discrete-based representations for modeling the emission probabilities, a consistent method was necessary to represent the continuous spatio-temporal data as discrete labels, according to the proposed task (i.e., symbols that represent visits occurring in a specific area).

The Hierarchical Triangular Mesh (HTM) is a method for dividing a spherical surface, such as the Earth, into triangular regions (i.e., trixels) of a form and size that is roughly uniform (Szalay et al., 2006).

**Figure 3.9:** Recursive triangular division of a sphere through a hierarchical triangular mesh.
In a very efficient way, this method provides the opportunity to index localized objects in the surface of a sphere (e.g., locations corresponding to coordinates), see Figure 3.9.

This method is configurable through a resolution parameter, which indicates the number of division iterations that should be considered. Each triangular shape can be divided into 4 new ones, through the midpoints of its edges. Initially, the sphere is divided into 8 triangular shapes (e.g., 4 shapes for each hemisphere). Thus, after the first iteration, we will have $8 \times 4^1 = 32$ triangular shapes, while after the second one we will have $8 \times 4^2 = 128$ regions and so on. Therefore, a resolution parameter $r$ can be used to control the number of trixels accordingly to the formula $8 \times 4^r$.

Given that this work focuses on the semantic classification of locations within human trajectories, we believe that considering regions of approximately 740 to 1500 m$^2$ would be appropriate for our task. Since the considered semantic classes include shops or entertainment regions, we considered a resolution of parameter 18 for the HTM approach, which generates triangles with the aforementioned characteristic. Then, with this method, we represent a geoposition (i.e., coordinates of latitude and longitude) through the HTM code of the corresponding triangular region. This $HTM_{code}$ constitutes the geospatial information present in the symbols used by our HMM models.

However, geospatial information does not completely reflect the distinct activities that can occur on the location under consideration. In some cases it is better to rely on temporal information to semantically classify a location. For instance, if in one location there are some shops, an art gallery and a nightclub, you can go shopping or visit an exhibition during the day, and later go to the night club. However, by only considering the geospatial information, we will not be able to classify semantically the activities performed at this location. With temporal information, it is easier to understand which activity occurred, since art galleries usually do not open every day, the same happening with clubs that, normally, receive visits during the night.
We therefore define our model’s symbols as the concatenation of both the geospatial and temporal information of each visit within a trajectory. Particularly, we consider the day of the week and the hour of the day to be the temporal marks that are more important for this task. The reason behind this choice is based on the fact that people usually make some activities according to some specific hours, or according to the day of the week, as described in the previous example. Henceforward, and as illustrated in Figure 3.10, the symbols used in our HMM models can be seen as tuples \(< \text{HTM code}, \text{Day of the week}, \text{Hour of the day} >\), with the day of the week mapped into a number from 1 to 7 (i.e., from Sunday to Saturday) and with the hour of the day mapped to a number from 0 to 23.

### 3.3.2 Using HMMs in a Supervised Setting

One particular problem with the usage of HMMs concerns with learning the model parameters. Considering a supervised setting, one can use annotated data obtained through the proposed heuristic trajectory classification method (i.e., using categories as states) to estimate the beginning, transition and emission probabilities. Consider that the parameters \(c(i \to j)\) and \(c(i \uparrow x)\) represent the number of existing transitions from state \(i\) to \(j\), and the number of times that symbol \(x\) is observed under state \(i\). After the counting process, and to avoid excluding any possibility (i.e., to avoid the appearance of cases with a probability of zero), we can apply Laplace smoothing to the estimation of the model parameters. As Equation 3.13 shows, in the case of the initial probabilities, we add 1 to the count of each beginning state, which represents the context of the first visit, and then we sum to the denominator the total number of states.

\[
\pi_{yi} = \frac{c_{t1}(i \to j) + 1}{\sum_{s \in Y} c_{t1}(i \to s) + |Y|} \quad (3.13)
\]

In Equation 3.14, for each transition we also add 1, but this time we add to the denominator the number of possible transitions to any state (i.e., \(|Y|\)).

\[
a_{yi,yj} = \frac{c(i \to j) + 1}{\sum_{s \in Y} c(i \to s) + |Y|} \quad (3.14)
\]

Finally, Equation 3.15 presents the smoothing procedure over the emission probabilities. Here, we add 1 to the relative count on a symbol under a given state, and we later add to the division the total number of symbols (i.e., \(|X|\)).

\[
b_{yi,x} = \frac{c(i \uparrow x) + 1}{\sum_{\rho \in X} c(i \uparrow \rho) + |X|} \quad (3.15)
\]
To circumvent the problem of sparse data, we also designed a process to smooth the emission probabilities based on the HTM method. In this procedure, we use two supervised HMMs with distinct geospatial resolutions, where a coarse-grained model smooths a thin-grained one (i.e., the model with a larger number symbols and a higher geospatial resolution). The rational for the implementation of this smoothing strategy is that, for a given timestamp, we can obtain the activities’ distribution (i.e., the context for the visits) over a larger area, and use it as a component for the activities’ distribution of each smaller area within the larger one. In this way, we can construct a probability distribution on the context of visits to symbols that do not appear in the training set. This does not happen when we use only the Laplace smoothing procedure, since for these specific symbols, in the context of a given state, the emission probability will be constant. The following equation shows the strategy used for smoothing the emission probabilities:

\[
b_{y_i,x_t} = (b_{y_i,x_t} \times 0.8) + \left( \frac{b_{y_i,z_t} + 1}{|Z|} \times 0.2 \right)
\]

This equation has two components, one considering the Laplace smoothing with a weight of 80%, and another representing the smoothing factor of the strategy described before, and with a weight of 20%. The \(b_{y_i,z_t}\) parameter is the emission probability of a coarse-grained symbol in a given state \(y_i\). Lastly, both \(X\) and \(Z\) respectively correspond to the sets of symbols of the thin-grained and coarse-grained models. Note that we consider only symbols of the dataset. If a larger area contains smaller ones that do not appear in the dataset, then we distribute its probability mass between the existing areas. Another consideration is that, as Equation 3.15 shows, we only use Laplace smoothing in the coarse-grained model. In order to be able to relate symbols of distinct resolutions, but that share the same timestamp and their intersecting areas, we use the hierarchical structure of the HTM codes, which contain the number of thin-grained symbols that compose a coarse-grained symbol (i.e., this is variable, since we consider only symbols that occur in the dataset).

### 3.3.3 Using HMMs in an Unsupervised Setting

Another approach for addressing the semantic classification of locations, using HMMs, involves the usage of an unsupervised procedure for finding the model parameters. In opposition to the supervised approach, this one does not require any contextual information (e.g., semantic annotations), thus being independent of the heuristic solution. More specifically, given a set of trajectories, only the symbols that represent the spatio-temporal locations of each trajectory are used to support the task of learning the parameters. The goal is to find the parameters that best
explain a given set of sequences.

There are several well-known algorithms that can derive efficiently a local maximum likelihood HMM model, such as Baum-Welch. Several works use this algorithm, recognizing it as being very effective in general, resulting in accurate models (Allahverdyan & Galstyan, 2011; Rodríguez & Torres, 2003). The Baum-Welch algorithm is a special case of the expectation-maximization (EM) algorithm, which uses the principle of dynamic programming through the forward-backward algorithm for estimating the HMM parameters.

For describing this algorithm, we consider an HMM model \( \lambda = < A, B, \pi > \) with the notation presented earlier, within the context of having \( N \) possible states and \( S \) different possible symbols (Dugad & Desai, 1996). Here, \( A \) is a matrix with the time-independent transition probabilities between states, \( B \) is a matrix that contains the emission probabilities corresponding to the observations of a particular symbol given a state, and \( \pi \) is a vector with the initial state probabilities, which is the probability that represents the states observed in the first position of the trajectory. In order to approximate the supervised setting for future comparisons, we configured our unsupervised model with \( 9 \) possible states (i.e., \( N = 9 \)).

In particular, the Baum-Welch algorithm aims to find \( \lambda^* = \max \lambda P(X|\lambda) \), where \( X \) can be a set of trajectories or one given trajectory. Let us consider \( x = < x_1, x_2, ..., x_T > \) as a trajectory with \( T \) locations. This algorithm starts by initializing the HMM’s parameters with random values and, iteratively, updates them until convergence, or up until reaching a configurable number of steps.

An expectation step computes the probabilities for each state that possibly completes the trajectory’s location information, using the current parameters of \( \lambda \) and the Forward-Backward algorithm. In brief, we have that the Forward algorithm calculates \( \alpha_i(t) = P(x_1, \ldots, x_t, Y(t) = y_i|\lambda) \), which is the probability of seeing the partial observable sequence \( x_1, \ldots, x_t \) ending up in state \( y_i \) at time \( t \). Efficiently, we can get these probabilities by using the following equations:

\[
\alpha_{y_i}(1) = \pi_{y_i} b_{y_i, x_1}
\]

\[
\alpha_{y_i}(t+1) = b_{y_i, x_{t+1}} \sum_{j=1}^{N} \alpha_{y_j}(t) \times a_{y_j, y_i}
\]

Consider \( a_{y_j, y_i} \) as the transition probabilities and \( b_{y_i, x_t} \) as the emission probabilities in the previous equations. Later, the Backward algorithm calculates the probability of the remaining and ending part of the sequence \( x_{t+1}, \ldots, x_T \), given that we start at position \( t \) with the state \( y_i \). Similarly to the Forward algorithm, this step calculates recursively the values of \( \beta_{y_i}(t) \) through the equations:
\[ \beta_{y_i}(T) = 1 \]  
\[ \beta_{y_i}(t) = \sum_{j=1}^{N} \beta_{y_j}(t+1) \alpha_{y_i,y_j} b_{y_j,x_{t+1}} \]  

With the values of \( \alpha_{y_i} \) and \( \beta_{y_i} \), we are able to calculate \( \gamma_{y_i} \), which is the probability of a particular state at position \( t \), and \( \xi_{y_i,y_j} \), which is the transition probability from a state to another in a specific position \( t \).

\[ \gamma_{y_i}(t) \equiv P(Y(t) = y_i | X, \lambda) = \frac{\alpha_{y_i}(t) \beta_{y_i}(t)}{\sum_{j=1}^{N} \alpha_{y_j}(t) \beta_{y_j}(t)} \]  
\[ \xi_{y_i,y_j}(t) \equiv P(Y(t) = y_i, Y(t+1) = y_j | X, \lambda) = \frac{\alpha_{y_i}(t) \alpha_{y_i,y_j} \beta_{y_j}(t+1) b_{y_j,x_{t+1}}}{\sum_{i=1}^{N} \sum_{j=1}^{N} \alpha_{y_i}(t) \alpha_{y_i,y_j} \beta_{y_j}(t+1) b_{y_j,x_{t+1}}} \]  

A maximization step can then re-estimate the model's parameters using the values of \( \gamma \) and \( \xi \). The equations shown below correspond to the update rules:

\[ \pi_{y_i} = \gamma_{y_i}(1) \]  
\[ a_{y_i,y_j} = \frac{\sum_{t=1}^{T-1} \xi_{y_i,y_j}(t)}{\sum_{t=1}^{T} \gamma_{y_i}(t)} \]  
\[ b_{y_i,x_k} = \frac{\sum_{t=1}^{T} \delta_{x_t,x_k} \gamma_{y_i}(t)}{\sum_{t=1}^{T} \gamma_{y_i}(t)} \]  

In the nominator of the equation that gives \( b_{y_i,x_k} \), the summation is only made over observed symbols equal to \( x_k \), that is, \( \delta_{x_t,x_k} = 1 \) if \( x_t = x_k \) and \( \delta_{x_t,x_k} = 0 \) otherwise. Until convergence and after updating the parameters \( \pi \), \( A \) and \( B \), the Baum-Welch algorithm executes a new iteration of the presented procedure.

Due to the characteristics of our trajectory dataset, and with the purpose of improving the model’s accuracy, we apply a similar smoothing strategy to the one described in Section 3.3.2. More specifically, consider another model with symbols of a low HTM resolution. Each one of these symbols relates to one or more higher resolution symbols, since they represent a larger area which includes at least one of them. The estimation of parameters in the lower resolution model occurs in the same way as in the original one, using the algorithm described earlier. The difference lies in the maximization step, where to re-estimate the emission probability \( b_{y_i,x_k} \) for a given higher resolution symbol \( x_k \), we consider the emission probability of the corresponding lower
resolution symbol $r_{ui}$, as follows:

$$b_{yi,x_k} = (0.8 \times b_{yi,x_k}) + (0.2 \times b_{yi,x_u})$$  \hfill (3.26)

### 3.4 Summary

In this chapter, I detailed the considered approaches to address the problem of the semantic classification of locations within trajectories. I started by presenting the raw data properties, the required behavioral information, and the questions I planned to address through my research work. Then, I described each proposed classification method. I began by exposing the heuristic classification approach that includes some common sense assumptions about daily activities, and which uses the semantic information from a LBSN. Next, I introduced a classification approach using HMMs, where the HTM method is used to overcome some challenges regarding trajectory data. Finally, for each HMM training procedure (i.e., supervised and unsupervised), I detailed the formulae used to build the models. It is important to mention that these formulae incorporate strategies to minimize the effect of sparse data on the learning process, namely through model smoothing.
Chapter 4

Validation Experiments

This chapter presents the results of the experiments executed to validate the proposed approaches. I start by providing a succinct characterization of the trajectory dataset used in these experiments. Then, the evaluation methodology is presented, followed by a formal presentation of the metrics used to assess the quality of the obtained results. Finally, the results for each experiment are presented and further discussed. These results relate to experiments concerning with semantically classifying locations in human trajectories using HMMs, considering areas of different sizes and distinct learning approaches, namely supervised and unsupervised.

4.1 The GeoLife Dataset

The dataset used for testing the proposed approaches is the Microsoft Research GeoLife GPS Trajectories dataset (Zheng et al., 2008, 2009, 2010). This dataset is characterized by having been mostly collected in China (i.e., in over 30 cities, within an emphasis on Beijing), although also in some European and American cities, for a period of 4 years (i.e., from April 2007 to October 2011) by 178 users. These data contain 17,621 GPS trajectories of locations visited in sequence, making for a total distance of about 1.2 million kilometers and a total duration of over 48,000 hours. The GPS locations of the trajectories within the dataset were captured every 15 seconds or every 510 meters, and include routine movements as well as leisure activities.

After enriching the GeoLife dataset with contextual information resulting from an heuristic supported on data collected from Foursquare, there are still locations where the contextual information is unknown. Given the objective to view each location as a staying point, where people
perform some activity, we decided to exclude locations without contextual information. For training a model it is important to have a reasonable sample of trajectories. Due to sparse data, if we exclude all the trajectories that include locations with a point corresponding to an unknown context, the trajectory sample is considerably reduced. Therefore, regarding the trajectories that include locations with an unknown context, we decided to use parts of these trajectories, that can still be considered a trajectory (i.e., if these parts contain at least two locations).

Due to the way the data were captured, another problem identified on the dataset concerns with the repeated locations within a given trajectory. Thus, we filter redundant locations through the combination of two techniques, which use the angle of direction, time, and distance measurements. The first technique leverages on the intuition that it is likely for a person to change its activity when changing the direction of the trajectory. So, as Figure 4.11 shows, we pick each 3 locations windows within a particular trajectory to evaluate the middle one, and check if there is a change in direction. Considering that the locations within the location window respect a time-series, the change of direction is identified when the angle, formed between the first location and the last one, is outside the threshold of $[160^\circ$ to $200^\circ]$.

With this threshold, we try to include possible GPS inaccuracies and identify if the 3 locations within the trajectory have a consensual direction by drawing them as a vector with two segments. More specifically, when decomposing the vector’s segments, there are at least two components from the distinct segments that agree in terms of direction. We use Vincenty’s formulae (Vincenty (1975)) with the location’s coordinates to calculate the distances between points and then, by knowing the distances, it is possible to know the location’s angle through the Law of Cosines\(^1\).

If the angle value is outside of the threshold, we add the location under evaluation to the clean trajectory, since we consider that a change of direction has occurred and possibly also a change of activity. Otherwise the locations agree in terms of direction and, considering the way the locations were collected for this dataset (i.e., every 15 seconds), we perform further analyses to

verify if there was an activity change between the first location and the one under evaluation. On this analysis, we start by using a technique that checks for two conditions so we can add the location under evaluation to the clean trajectory. One is if the location under evaluation has a different HTM code than the first location. The other is if they have distinct categories. In addition, the location under evaluation is automatically added to the clean trajectory, if it is the first or the last one of a given trajectory, apart of its known context.

For carrying out the trajectory classification task, we experimented with different geospatial configurations. Table 4.3 presents the dataset’s characteristics under four different geospatial configurations, from where we can see the influence of the geospatial resolution parameter on the enriched dataset (i.e., GeoLife locations associated to a category from Foursquare). The first segment is about the trajectory samples and the locations within them. We can observe that the number of trajectories is the same for the different geospatial parameters, but the number of locations within these trajectories varies. The higher the geospatial resolution parameter, the more locations within trajectories do exist. The increase of the average and standard deviation, relative to the locations per trajectory, relates directly with this. The second segment tells us how many HTM trixels we used to index the locations of our trajectories, together with the main characteristics of these trixels, namely the area and the positioning range, in our corresponding geospatial resolution models. The third segment shows that the timestamps associated to the dataset remain constant for all characterizations, as the geospatial parameter does not influence this information. Finally, the fourth segment presents the number of considered symbols in each case. The geospatial resolution influences these values, since they result from the combination of trixels with temporal marks. Table 4.3 provides an overview on the difficulty associated with the classification task, since it confirms that the data are sparse due to the huge amount of symbols available for each experiment. This fact has a direct effect on model complexity, which is directly proportional to the amount of considered symbols.
4.2 Evaluation Methodology

For assessing the quality of the results obtained by the different approaches under consideration, we followed other related works or previous joint evaluation events (e.g., the Nokia Mobile Data Challenge), in terms of evaluation guidelines and metrics.

Since the objective is to attribute semantic meaning to trajectories, we concentrate the evaluation of our models on the main entities supporting this task, namely the locations, the categories and the trajectories themselves as a whole. In this way, we assessed the model’s performance from two perspectives, one having locations and the other having categories as the subject of analysis. The locations perspective is only concerned with the model’s classification performance regarding locations and trajectories, rather than discriminating what is the semantic information associated with each of them. The categories perspective filled this evaluation space (i.e., categories were converted into the states of the HMM), where the model’s performance is analyzed for each category.

Note that unsupervised models do not take into account the semantic information (i.e., the categories) directly. Therefore, when estimating the models’ parameters, the algorithm learns how to classify a location with a specific state (i.e., one from a predetermined number of states) based only on its spatio-temporal features. In order to evaluate the unsupervised models through a category perspective, and to compare them with the supervised ones, we created a method to identify category-state pairs. These category-state pairs are identified through the combination of temporal features with a Root Mean Square Deviation measure. For further details about this challenge and about the evaluation of the unsupervised results, refer to Section 4.3.2.

Model evaluation was mostly based on the standard metrics of precision, recall and F1.

**Precision** is the fraction of classified elements that are relevant to one given class. More specifically, the precision for a class is the number of correctly classified elements (i.e., the number of items well labeled as belonging to the class in question) divided by the total number of elements labeled as belonging to this class. This is represented in Equation 4.27.

\[
\text{precision}(A) = \frac{|\text{items belonging to class } A \cap \text{items classified as } A|}{|\text{items classified as } A|} \quad (4.27)
\]

**Recall** is the fraction of relevant elements that are classified to one given class, which translates to the number of correctly classified elements divided by the total number of elements that actually belong to this specific class, which includes elements wrongly classified as belonging to another class. This can be seen in Equation 4.28.
4.2. EVALUATION METHODOLOGY

\[
\text{recall}(A) = \frac{|\text{items belonging to class } A \cap \text{items classified as } A|}{|\text{items belonging to class } A|}
\]

(4.28)

\( \textbf{F1} \) is a measure of a classifier’s accuracy that combines precision and recall by calculating the harmonic mean of them. This is given by Equation 4.29.

\[
\text{F1} = 2 \times \frac{\text{precision} \times \text{recall}}{\text{precision} + \text{recall}}
\]

(4.29)

The above metrics only consider classification quality under the context of a single existing class, although my classification problem includes several possible classes (i.e., nine semantic categories) for the locations. In order to be able to determine the overall classification performance, I used two additional approaches, namely the \textit{micro} and \textit{macro} averages over the previously defined evaluation metrics.

\textbf{Micro Averages} consider that all classified elements have the same weight, despite the class to which they belong.

\[
\text{precision}_{\text{micro}} = \frac{\sum_{i=0}^{|C|} |\text{items belonging to class } i \cap \text{items classified as } i|}{\sum_{i=0}^{|C|} |\text{items classified as } i|}
\]

(4.30)

\[
\text{recall}_{\text{micro}} = \frac{\sum_{i=0}^{|C|} |\text{items belonging to class } i \cap \text{items classified as } i|}{\sum_{i=0}^{|C|} |\text{total items belonging to class } i|}
\]

(4.31)

Notice that in the above Equations, an equal weight is given to each element being classified, and therefore we can have that larger classes have more influence on the results.

\textbf{Macro Averages}, on the other hand, consider that all the classes have the same weight, regardless of how many elements belong to them.

\[
\text{precision}_{\text{macro}} = \frac{1}{|C|} \sum_{i=0}^{|C|} \frac{|\text{items belonging to class } i \cap \text{items classified as } i|}{|\text{items classified as } i|}
\]

(4.32)

\[
\text{recall}_{\text{macro}} = \frac{1}{|C|} \sum_{i=0}^{|C|} \frac{|\text{items belonging to class } i \cap \text{items classified as } i|}{|\text{total items belonging to class } i|}
\]

(4.33)

With macro averages, the results can be more reliable when classes do not have the same number of elements.
**Accuracy** is yet another metric that can be used in the context of multi-class problems. Its value is given by the number of well classified elements, divided by the total number of executed classifications.

\[
\text{accuracy} = \frac{|\text{correctly classified items}|}{|\text{total classified items}|} 
\] (4.34)

I will perform classification experiments with the dataset described in Table 4.3, using the previous metrics to access the quality of the results.

### 4.3 Experimental Results

The section presents the results obtained from the conducted experiments, together with their discussion. In the first place, the results of the experiments for the semantic classification of locations in human trajectories using supervised HMMs are presented and discussed. We assess the quality of these results from both the locations and the categories perspectives. These results then are further discussed, and the section also provides a deeper exploration of the model that obtained the better results.

Later, the results for experiments using unsupervised HMMs to perform the semantic classification of locations in human trajectories are presented and discussed. This section also presents a method that, by matching each unsupervised state to a category, allows the direct comparison of these results with the ones obtained with supervised models.

#### 4.3.1 Results with Supervised HMMs

<table>
<thead>
<tr>
<th>Geospatial Resolution</th>
<th>16</th>
<th>16(_{10})</th>
<th>18</th>
<th>18(_{10})</th>
<th>19</th>
<th>19(_{10})</th>
</tr>
</thead>
<tbody>
<tr>
<td>Accuracy (%)</td>
<td>42.9</td>
<td>48.2</td>
<td>43.3</td>
<td>56.5</td>
<td>36.9</td>
<td>52.5</td>
</tr>
<tr>
<td>Micro Average</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Precision (%)</td>
<td>42.9</td>
<td>48.2</td>
<td>43.3</td>
<td>56.5</td>
<td>36.4</td>
<td>52.6</td>
</tr>
<tr>
<td>Recall (%)</td>
<td>42.9</td>
<td>48.2</td>
<td>43.3</td>
<td>56.5</td>
<td>36.4</td>
<td>52.6</td>
</tr>
<tr>
<td>Macro Average</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Precision (%)</td>
<td>45.0</td>
<td>42.6</td>
<td>54.6</td>
<td>49.8</td>
<td>56.4</td>
<td>47.5</td>
</tr>
<tr>
<td>Recall (%)</td>
<td>39.0</td>
<td>51.5</td>
<td>42.7</td>
<td>56.9</td>
<td>39.1</td>
<td>52.9</td>
</tr>
<tr>
<td>=100%</td>
<td>30.7</td>
<td>32.2</td>
<td>24.4</td>
<td>32.7</td>
<td>26.1</td>
<td>29.1</td>
</tr>
<tr>
<td>Correctly Classified (%)</td>
<td>≥ 90%</td>
<td>31.5</td>
<td>33.0</td>
<td>32.2</td>
<td>38.3</td>
<td>29.0</td>
</tr>
<tr>
<td>Trajectories (%)</td>
<td>≥ 80%</td>
<td>36.3</td>
<td>39.5</td>
<td>37.8</td>
<td>46.9</td>
<td>33.4</td>
</tr>
<tr>
<td></td>
<td>≥ 70%</td>
<td>41.5</td>
<td>45.9</td>
<td>43.2</td>
<td>52.7</td>
<td>37.4</td>
</tr>
</tbody>
</table>

*Table 4.4: Results of different supervised experiments.*
4.3. EXPERIMENTAL RESULTS

Table 4.5: Results for each of the individual HMM states.

<table>
<thead>
<tr>
<th>Geospatial Resolution</th>
<th>16</th>
<th>16_{10}</th>
<th>18</th>
<th>18_{10}</th>
<th>19</th>
<th>19_{10}</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Nightlife Spot</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Precision (%)</td>
<td>49.5</td>
<td>18.7</td>
<td>70.6</td>
<td>17.5</td>
<td>73.5</td>
<td>13.9</td>
</tr>
<tr>
<td>Recall (%)</td>
<td>11.0</td>
<td>53.8</td>
<td>18.8</td>
<td>55.1</td>
<td>19.2</td>
<td>51.1</td>
</tr>
<tr>
<td>F1 (%)</td>
<td>18.0</td>
<td>27.8</td>
<td>29.7</td>
<td>26.6</td>
<td>30.4</td>
<td>21.8</td>
</tr>
<tr>
<td><strong>Arts and Entertainment</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Precision (%)</td>
<td>47.3</td>
<td>36.3</td>
<td>55.7</td>
<td>42.8</td>
<td>53.3</td>
<td>38.9</td>
</tr>
<tr>
<td>Recall (%)</td>
<td>31.8</td>
<td>50.2</td>
<td>32.4</td>
<td>51.2</td>
<td>28.6</td>
<td>44.3</td>
</tr>
<tr>
<td>F1 (%)</td>
<td>37.3</td>
<td>42.1</td>
<td>41.0</td>
<td>46.6</td>
<td>37.2</td>
<td>41.4</td>
</tr>
<tr>
<td><strong>Shop and Services</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Precision (%)</td>
<td>54.8</td>
<td>51.1</td>
<td>69.2</td>
<td>64.7</td>
<td>75.3</td>
<td>64.6</td>
</tr>
<tr>
<td>Recall (%)</td>
<td>34.2</td>
<td>38.5</td>
<td>37.2</td>
<td>51.2</td>
<td>30.9</td>
<td>47.8</td>
</tr>
<tr>
<td>F1 (%)</td>
<td>42.1</td>
<td>43.9</td>
<td>48.4</td>
<td>57.2</td>
<td>43.8</td>
<td>54.9</td>
</tr>
<tr>
<td><strong>Food</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Precision (%)</td>
<td>61.3</td>
<td>61.5</td>
<td>74.3</td>
<td>71.1</td>
<td>78.3</td>
<td>70.1</td>
</tr>
<tr>
<td>Recall (%)</td>
<td>45.6</td>
<td>46.1</td>
<td>40.6</td>
<td>56.5</td>
<td>31.1</td>
<td>52.8</td>
</tr>
<tr>
<td>F1 (%)</td>
<td>52.3</td>
<td>52.7</td>
<td>52.5</td>
<td>63.0</td>
<td>44.5</td>
<td>60.2</td>
</tr>
<tr>
<td><strong>Outdoors and Recreation</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Precision (%)</td>
<td>27.9</td>
<td>38.6</td>
<td>9.4</td>
<td>50.8</td>
<td>7.6</td>
<td>49.9</td>
</tr>
<tr>
<td>Recall (%)</td>
<td>42.3</td>
<td>58.0</td>
<td>67.5</td>
<td>58.3</td>
<td>75.1</td>
<td>53.0</td>
</tr>
<tr>
<td>F1 (%)</td>
<td>33.6</td>
<td>46.3</td>
<td>16.5</td>
<td>54.3</td>
<td>13.7</td>
<td>51.4</td>
</tr>
<tr>
<td><strong>College and University</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Precision (%)</td>
<td>22.8</td>
<td>44.3</td>
<td>40.2</td>
<td>42.0</td>
<td>43.4</td>
<td>34.4</td>
</tr>
<tr>
<td>Recall (%)</td>
<td>81.0</td>
<td>76.5</td>
<td>72.6</td>
<td>79.1</td>
<td>66.4</td>
<td>78.4</td>
</tr>
<tr>
<td>F1 (%)</td>
<td>35.6</td>
<td>56.1</td>
<td>51.8</td>
<td>54.9</td>
<td>52.5</td>
<td>47.8</td>
</tr>
<tr>
<td><strong>Residence</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Precision (%)</td>
<td>37.3</td>
<td>29.2</td>
<td>49.8</td>
<td>31.0</td>
<td>49.3</td>
<td>25.1</td>
</tr>
<tr>
<td>Recall (%)</td>
<td>30.3</td>
<td>52.5</td>
<td>33.2</td>
<td>54.7</td>
<td>30.0</td>
<td>50.8</td>
</tr>
<tr>
<td>F1 (%)</td>
<td>33.5</td>
<td>37.5</td>
<td>39.8</td>
<td>39.6</td>
<td>37.3</td>
<td>33.6</td>
</tr>
<tr>
<td><strong>Professional and Other Places</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Precision (%)</td>
<td>54.1</td>
<td>50.7</td>
<td>67.8</td>
<td>61.8</td>
<td>73.0</td>
<td>60.3</td>
</tr>
<tr>
<td>Recall (%)</td>
<td>36.1</td>
<td>42.8</td>
<td>36.3</td>
<td>51.4</td>
<td>30.1</td>
<td>48.6</td>
</tr>
<tr>
<td>F1 (%)</td>
<td>43.3</td>
<td>46.4</td>
<td>47.3</td>
<td>56.1</td>
<td>42.6</td>
<td>53.9</td>
</tr>
<tr>
<td><strong>Travel and Transport</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Precision (%)</td>
<td>50.2</td>
<td>53.1</td>
<td>54.3</td>
<td>69.9</td>
<td>53.6</td>
<td>70.3</td>
</tr>
<tr>
<td>Recall (%)</td>
<td>39.3</td>
<td>45.5</td>
<td>46.0</td>
<td>54.8</td>
<td>40.2</td>
<td>48.9</td>
</tr>
<tr>
<td>F1 (%)</td>
<td>44.1</td>
<td>49.0</td>
<td>49.8</td>
<td>60.3</td>
<td>46.0</td>
<td>57.7</td>
</tr>
</tbody>
</table>

We experimented with six distinct supervised HMMs, two for each data resolution from Table 4.3. Table 4.4 shows the overall results for these experiments, where we use the geospatial resolution parameter to describe the models. For representing the models that include the smoothing strategy relying on a coarse-grained model, we use two values where the smaller one (i.e., 10) is the geospatial resolution parameter of the coarse-grained HMM. As a first and interesting observation over the results, we can report that the HMM with geospatial resolution of $18_{10}$ is the most accurate supervised model, and the HMM with geospatial resolution of 19 is the less accurate one.

Another interesting observation is that all models that included the more sophisticated smoothing strategy performed better than the ones without it (i.e., in terms of accuracy, $18_{10} > 19_{10} > 16_{10} > 18 > 16 > 19$). Table 4.4, and also Table 4.3, confirms an expected correlation between the geospatial resolution and the accuracy. Considering the existing categories, it is very likely to have visits to small locations (e.g., shops or restaurants) and we correctly expected that a model
whose range can include small or medium areas will perform better. More specifically, as Table 4.3 shows, the area with geospatial resolution of 18 is the one that better fits this scenario, since it can include a regular small business location with its area ranging from 740 to 1534 \( m^2 \). Table 4.3 confirms the correlation by showing that, when considering distinct smoothing strategies, the best models are \( 18_{10} \) and 18. Other models do not possess the same capability, because they either look upon very large or very small locations.

In a different perspective, Table 4.5 details the results obtained for each state, each one corresponding to one semantic category. The first interesting information that we notice from this table is that the most accurate result corresponds to the Food state in the model \( 18_{10} \). In opposition, we observe that the less accurate state is the Outdoors and Recreation state in the model \( 19 \). Concerning the fact that Food is the most accurate state, we have that this result is expected since the majority of the locations were labelled with this category. Two observations prove this statement, one in the mentioned table, where the Food state is the most successful one in four different models and is very close to the ones in the remaining experiments, and another in Figure 4.12, where we show the amount of locations by category present in the dataset filtered with geospatial parameter of 18. Regarding state Outdoors and Recreation having the worst overall result, we were not surprised due to the fact that this category could imply locations with huge areas or where the area varies significantly. Also, Figure 4.12 indicates that, in the dataset, there
4.3. EXPERIMENTAL RESULTS

are few locations labeled with this category when compared to the other ones. Still in this perspective, it is interesting to report that the Nightlife Spot state achieved the worst results on four models, which it can be explained through the same reasons pointed out when discussing the Outdoors and Recreation state.

We also observe that model $18_{10}$ achieved the majority of the best results by state, contrasting with model $16$ that, with the exception of the Food and Outdoors and Recreation states, obtained the worst ones. Another curious observation is the improvement obtained from the smoothing strategy concerning the models with geospatial resolution of 18 and 19, in the context of the Outdoors and Recreation state. This is the perfect example of how the smoothing strategy helps to highlight smaller areas which probably compose the same location.

An interesting pattern observed in our results is the existing inverse relation between our smoothing strategy and the geospatial resolution, in which our smoothing strategy loses effectiveness when decreasing the considered area’s size. Table 4.5 exposes this rational through the F1 measure improvement within models of equal geospatial resolution. Almost all states improve for the models with model considering a geospatial resolution of 16, while two states fail to improve on the models with a geospatial resolution of 18, and another three fail to improve on the remaining models. Finally, we identify an interesting pattern when comparing models with the same smoothing approach, but with distinct geospatial resolutions. This pattern is the increase of the F1 measure when comparing a 16 model to a 18 one, and later, a decrease of the same measure when comparing a 18 model to a 19 one. The states not included in this pattern are the ones that have big differences in terms of the F1 measure between models with the same geospatial resolution (e.g., for the models with geospatial resolution of 18, the difference in the F1 measure is 38% on the state Outdoors and Recreation), namely Nightlife Spot, Outdoors and Recreation and College and University. The classification difficulty associated to these states lies with the features present in the considered symbols and on the amount of them. When considering these activities, we expect a sparse existence of multiple locations or different timestamps appearing in our trajectories.

From the obtained results, we conclude that the best model is $18_{10}$. Therefore, we will explore in detail several elements of this model, and occasionally from other ones for addressing the proposed questions. Table 4.6 establishes a more specific evaluation context by presenting a detailed description of the training and test sets for the model $18_{10}$. In this table, we divided the information items in the same manner as in Table 4.3. When looking at the first segment, we can observe that in both sets there are a huge number of distinct locations, leading the model to consider a huge amount of symbols. Another interesting information supporting the initial affirmation, and consistent for both sets, it is the different number of locations that can be on a
trajectory and the high average value of the number of locations by trajectory. Moving on to the
next segment, we observe data related to the geospatial resolution. Here, as a consequence
of the locations considered, we have a huge sum of HTM trixels considered by this model in
both sets. The third segment shows the timestamp period considered on both sets, which has
a difference of a few days. Temporal marks that we include in the symbols respect these period
boundaries. Lastly, we have a huge amount of symbols used for training and testing, since they
are a combination of HTM trixels with temporal marks. The most interesting information in Table
4.6 is the percentage of the new HMM symbols in the test set. This percentage implies that the
model has to classify nearly 18% of locations that were not present in the training trajectories.

Besides presenting results related to classification accuracy, we can also analyze other elements,
such as the categories that are typically visited in sequence within the trajectories. Figure 4.13
illustrates the transition probabilities between states. Considering the small average number of
categories by trajectory, together with the fact that we had a large chunk of trajectories where
there exists only one category, the question of popular transitions between activities could not be
addressed in a clear form. Transitions between different types of activities were rarely found.

Figures 4.14 and 4.15 address the question about which categories are most popular, at par-
ticular geographic regions or at particular temporal intervals. The bar plots are organized in a
top-down approach concerning the maximum number of visits by category, where each bar plot
illustrates the results for a corresponding category. Note that, in each bar plot, the dark bars
represent the HMM classification results and the remaining colored bars represent the annotated
data in the heuristic classification approach. Therefore, Figure 4.14 presents the number of visits
per day, where Food is the most popular category and the less popular category is Residence.
4.3. EXPERIMENTAL RESULTS

Figure 4.13: Probabilities associated to the supervised model.

Figure 4.14 also shows two interesting behavior patterns, namely the peak of activity in the period between 9 am and 2 pm, and the less data available in the period from 3 pm to 10 pm.

Another pattern can be found involving the Nightlife Spot, Outdoors and Recreation and Arts and Entertainment category-state pairs. When observed, the classification peaks are comparable and that could indicate possible classification errors. Notice that the periods from 2 to 3 am and from 5 to 7 am present an equal pattern on Arts and Entertainment and on Nightlife Spot. Also, the Nightlife Spot, Outdoors and Recreation and Arts and Entertainment category-state pairs appear to be related, taking into account their major peaks. In addition to the aforementioned periods, there is a peak at 23 pm common among the previously mentioned three category-state pairs.

Regarding classification errors, as Figure 4.14 illustrates and as we can then confirm through Table 4.8, the model confused the category Professional and Other Places with College and University. Moreover, Table 4.8 reveals that Food visits have an important weight on classification errors, but a higher rate of incorrect classifications involves the Nightlife Spot, Residence and College and University categories.
4.3.2 Results with Unsupervised HMMs

When using an unsupervised approach, one does not consider the semantic information for training the model. This way, the categories have no direct relation with the states, and thus we cannot directly compare HMM states against categories, to evaluate the performance of our models.

In order to measure the performance of our unsupervised models, we choose to train them considering 9 different states for two obvious reasons. First, we want to be consistent and compare this approach with the supervised one, which considered an equal number of states due to the number of available categories. Second, by considering 9 states we can then build a relatively simple method that solves this problem by matching each category to one of the model’s states.

Realizing that there is a strong relation between the timestamp and the activity performed by a person, our matching method uses the considered time features (i.e., Day of the week and Hour of the day) to identify the category (i.e., the information that represents the context of an activity) that best represents a state’s classified locations.

Our method to identify the category-state pairs uses a two-step algorithm. Using the time features, the method first calculates all possible similarities between categories and states, and then proposes a map of category-state pairs.
Initially, we have 4 matrices containing the number of locations, that relate each considered time feature with the categories or states (e.g., State by Day of the week matrix). Our method uses the Root Mean Square Deviation measure to look for the similarity between a category and a state. With this measure, we compare each state to all categories and build a matrix containing insights into how, through the time analysis, the locations were previously annotated and how the unsupervised model classified them. In order to adapt the Root Mean Square Deviation measure to the time features, when evaluating the difference between a category $\hat{\text{cat}}$ and a state $\hat{\text{y}}$, we start by observing the respective Day of the week matrices. For each day, the two Day of the week matrices respectively contain the number of locations classified as belonging to a given state $\hat{\text{y}}_{\text{day}}$ and annotated with a certain category $\hat{\text{cat}}_{\text{day}}$. As Equation 4.35 shows, we calculate the difference associated with the days of the week for the given category $\text{cat}$ and $\text{y}$ state.

$$\text{RSMD}_{\text{Day of the week}} = \sqrt{\frac{\sum_{\text{day}=1}^{7} |\hat{\text{y}}_{\text{day}} - \hat{\text{cat}}_{\text{day}}|^2}{7}}$$

Regarding the two elements that we are evaluating and for normalization purposes, we also keep the maximum and minimum number of locations, respectively, the day_{max} and day_{min}. Afterwards, we repeat the same calculations for the hours of the day, as presented in Equation
Geospatial Resolution | 16 | 16\(_{10}\) | 18 | 18\(_{10}\) | 19 | 19\(_{10}\)
--- | --- | --- | --- | --- | --- | ---
Accuracy (%) | 21.6 | 24.1 | 21.7 | 33.5 | 20.0 | 19.0
Micro Average Precision (%) | 21.6 | 24.1 | 21.7 | 33.5 | 20.0 | 19.0
Recall (%) | 21.6 | 24.1 | 21.7 | 33.5 | 20.0 | 19.0
Macro Average Precision (%) | 14.2 | 11.7 | 11.7 | 4.7 | 7.5 | 9.2
Recall (%) | 11.8 | 11.5 | 11.3 | 11.1 | 11.0 | 11.0
Correctly ≫ 100% | 6.6 | 9.0 | 0.8 | 9.4 | 1.6 | 0.09
≥ 90% | 6.7 | 9.1 | 1.0 | 9.7 | 3.2 | 0.09
≥ 80% | 7.7 | 10.0 | 1.6 | 11.1 | 5.3 | 0.09
Trajectories | 9.4 | 10.9 | 2.2 | 12.9 | 6.4 | 0.09

Table 4.7: Results of different unsupervised experiments.

\[
\text{RSMD}_{\text{Hour of the day}} = \sqrt{\frac{\sum_{\text{hour}=1}^{24} |\hat{y}_{\text{hour}} - \hat{\text{cat}}_{\text{hour}}|^2}{24}}
\] (4.36)

Finally, as shown in Equation 4.37, we obtain the balanced difference between a category \( \text{cat} \) and a \( y \) state, which will be considered in the second step of our method.

\[
\text{Difference} = \left(\frac{\text{RSMD}_{\text{Day of the week}}_{\text{day max}} - \text{day min}}{0.5}\right) + \left(\frac{\text{RSMD}_{\text{Hour of the day}}_{\text{hour max}} - \text{hour min}}{0.5}\right)
\] (4.37)

After getting the differences between each state and all categories, and after building the matrix holding these values, our method identifies the matching state-category pairs. One starts by looking for the minimum value within the matrix. When the method finds this value, it identifies and associates as a pair, both the corresponding state and category. Until each state has a unique corresponding category, our method iterates over the remaining part of the matrix, that is, it does not consider both the line and the column corresponding to the elements that previously formed a pair. The rational behind this choice is to avoid cases, where two distinct categories could relate with the same state. This would happen due to the dataset’s characteristics, since the amount of locations associated with each category is very different.

Similarly to what was performed for the supervised approach, we experimented with six different unsupervised HMMs, and with two of them for each data characterization (i.e., geospatial configuration). Table 4.3.2 presents the obtained results for these experiments. Although the results from the unsupervised experiments had a worst performance when compared to the supervised ones, they confirm that the smoothing strategy works when considering larger areas in our models. In these results, we observe again that the HMM model with the geospatial resolution of \( 18_{10} \) is the most suited one, scoring 33.5% of accuracy, and the HMM with geospatial resolution...
### 4.4. SUMMARY

In this chapter, I presented and discussed the obtained results for the experiments focusing on the semantic classification of locations in human trajectories through HMMs. These experiments involved HMM models considering different area sizes, one smoothing strategy and two different training approaches, namely supervised and unsupervised.

For the HMM models using a supervised approach, we report that the model considering a geospatial resolution of $18_{10}$ is the most accurate one. This result confirmed our expectation that a model whose geospatial resolution can include small or medium areas will perform better in this type of task. Also, as expected, the Food category was the most accurate one, since the majority of the locations were labeled with this category. Moreover, the results confirmed the importance of the smoothing strategy, since it improved the accuracy of the models. Although $19_{10}$ is the less accurate one. The models that obtain worst results are the ones with higher geospatial resolution parameter (i.e., $19_{10}$).

![Table 4.8: Confusion matrix of the supervised model $18_{10}$ concerning the visit classifications (i.e., the proposed states) and their categories.](image)

<table>
<thead>
<tr>
<th>Categories</th>
<th>Nightlife Spot</th>
<th>Travel and Transport</th>
<th>Outdoors and Recreation</th>
<th>Shop and Services</th>
<th>College and University</th>
<th>Food</th>
<th>Arts and Entertainment</th>
<th>Residence</th>
<th>Professional and Other Places</th>
<th>Total</th>
</tr>
</thead>
<tbody>
<tr>
<td>Nightlife Spot</td>
<td>8 811</td>
<td>755</td>
<td>392</td>
<td>853</td>
<td>1 265</td>
<td>2 019</td>
<td>449</td>
<td>713</td>
<td>724</td>
<td>15 981</td>
</tr>
<tr>
<td>Travel and Transport</td>
<td>8 320</td>
<td>75 957</td>
<td>4 546</td>
<td>6 585</td>
<td>9 749</td>
<td>14875</td>
<td>5 008</td>
<td>5 089</td>
<td>8 365</td>
<td>138 494</td>
</tr>
<tr>
<td>Outdoors and Recreation</td>
<td>1 921</td>
<td>2 193</td>
<td>24 888</td>
<td>1 540</td>
<td>4 097</td>
<td>3 715</td>
<td>1 155</td>
<td>1 365</td>
<td>1 839</td>
<td>42 713</td>
</tr>
<tr>
<td>Shop and Services</td>
<td>6 803</td>
<td>6 458</td>
<td>4 110</td>
<td>68 058</td>
<td>13 952</td>
<td>17 596</td>
<td>3 547</td>
<td>5 354</td>
<td>7 008</td>
<td>132 886</td>
</tr>
<tr>
<td>College and University</td>
<td>956</td>
<td>1 313</td>
<td>969</td>
<td>1 878</td>
<td>64 269</td>
<td>6 957</td>
<td>1 092</td>
<td>1 500</td>
<td>2 313</td>
<td>81 247</td>
</tr>
<tr>
<td>Food</td>
<td>14 162</td>
<td>15 876</td>
<td>8 248</td>
<td>16 708</td>
<td>37 617</td>
<td>178 865</td>
<td>10 570</td>
<td>13 069</td>
<td>21 370</td>
<td>316 485</td>
</tr>
<tr>
<td>Arts and Entertainment</td>
<td>2 360</td>
<td>2 083</td>
<td>1 064</td>
<td>1 627</td>
<td>3 742</td>
<td>5 050</td>
<td>20 449</td>
<td>1 373</td>
<td>2 199</td>
<td>39 947</td>
</tr>
<tr>
<td>Residence</td>
<td>978</td>
<td>1 027</td>
<td>955</td>
<td>1 356</td>
<td>3 309</td>
<td>3 046</td>
<td>552</td>
<td>15 226</td>
<td>1 368</td>
<td>27 817</td>
</tr>
<tr>
<td>Professional and Other Places</td>
<td>6 020</td>
<td>7 897</td>
<td>3 774</td>
<td>6 618</td>
<td>14 947</td>
<td>19 383</td>
<td>5 010</td>
<td>5 419</td>
<td>72 961</td>
<td>142 029</td>
</tr>
<tr>
<td><strong>Total</strong></td>
<td>50 331</td>
<td>113 559</td>
<td>48 946</td>
<td>105 223</td>
<td>152 947</td>
<td>251 506</td>
<td>47 832</td>
<td>49 108</td>
<td>118 147</td>
<td>937 599</td>
</tr>
</tbody>
</table>

The models that obtain worst results are the ones with higher geospatial resolution parameter (i.e., $19_{10}$).
the results were inconclusive about the categories typically visited in sequence, we were able to determine that *Food* was the most popular category concerning visits per day. In addition, we found an interesting behaviour pattern, where the majority of the data was collected in the period between 9 am and 2 pm.

Regarding the experiments applying an unsupervised approach, the overall performance of the experimented HMM models was worst when comparing to the supervised approach. Nevertheless, we report that the most accurate model was again the one using a resolution of 18, and we again observed that the proposed smoothing strategy improves the results.
Chapter 5

Conclusions

In my MSc thesis, I proposed and compared different approaches for the semantic classification of visited locations within human trajectories. My approaches consist on a heuristic method, and two other methods based on Hidden Markov Models (HMMs), either relying on supervised or unsupervised settings. I also surveyed the current state-of-art of in terms of different techniques and methods to solve similar classification or prediction problems. From the experiments reported in this document, I can enumerate three main conclusions:

First, I argue that the GeoLife dataset is not directly suitable for the evaluation of this task, since data collection, instead of being directed to the activities, was temporally oriented (e.g., with devices collecting data every 5 seconds or 500 meters). Therefore, and specially for the unsupervised setting, for achieving more accurate results one would need more and different data collected with identical time features.

The most accurate model is the one whose area range is more diverse, in order to cover different locations where different activities can occur. Also, through the analysis of Figures 4.14 and 4.15, I realize that taking into account the temporal marks as unobservable parameters was not a relevant choice.

As a final conclusion, I can assert that for achieving good results in this classification task, there is no need for using highly sophisticated sequential data models. Once by adding a the novel smoothing strategy to a well-known approach, we have that the results improved 13.2% in the accuracy of the most accurate model, and we observed a global maximum of 15.6% on another model considering a smaller area size.
5.1 Summary of Results

The most important contributions of my MSc thesis are as follows:

HMMs as a solution for the semantic classification of the locations within human trajectories:
were proposed HMMs have been shown to be a good solution for the semantic classification
of the visited locations within human trajectories. When considering nine possible semantic
categories to classify a given location within a trajectory, through a series of experiments,
we measured an accuracy of 56% for the models trained using supervised learning, and 34% for
the models whose learning approach is an unsupervised one. In the context of the ob-
tained results and experiments, a novel smoothing strategy was applied to the HMM models
without making changes to their structure, and we showed that it is possible to meaning-
fully improve the results. More specifically, when the HMMs applied the novel smoothing
strategy, we observed an accuracy increase of 13.2% in the best supervised model, and a
global maximum of 15.6% in another supervised model considering smaller areas.

Implementation and adaptation of the HMM models: In order to conduct the study, and to
compare of distinct approaches concerning the semantic classification of locations in hu-
man trajectories, I implemented algorithms related to the HMM models. Since, the focus
was to study semantic classifications from several approaches, I implemented standard al-
gorithms for both supervised and unsupervised training. The novelty concerning the train-
ing algorithms was the addition of a smoothing strategy, which was used to overcome the
problem of sparse data. The novel smoothing strategy brings, to the HMM models, the as-
sumption that similar activities are usually concentrated in the same areas, and considering
this assumption, the algorithms attribute a different weight when normalising the activity
performed in a given location. For the unsupervised approach, I adapted the Baum-Welch
algorithm to include the novel smoothing strategy, and I implemented a custom algorithm
to map HMM states into semantic categories. The mapping was done using the root mean
square deviation measure combined with timestamp information of the locations (i.e., hour
of the day and day of the week). The considered timestamp information enabled the as-
seessment and comparison of the unsupervised approach with the supervised one. All the
source code was made available as an open-source project1, so that other researchers or
persons interested in this topic can use it:

1 https://code.google.com/p/alochist/
5.2. **FUTURE WORK**

**Data Enrichment package:** A procedure was implemented to collect data from Foursquare to enrich the GeoLife dataset. The source code adds to the spatio-temporal dataset the missing semantic information, retrieved in JSON objects. Also, an algorithm was built to associate to each location a category describing a possible activity in it. The category attribution was done considering Foursquare’s venues in a given location and some common sense assumptions. More specifically, I extracted, from the venues, the category of the closest venue, the most observed category among the close venues, and the category of the most visited venue.

5.2 Future Work

An interesting idea for the future work relates to addressing the same classification task with discriminative models, such as Conditional Random Fields (CRFs) or Support Vector Machines (SVM). I would also like to experiment with higher-order HMMs, which can take into account not only the previous state but multiple previous states.

Finally, I would also like to test our approaches with datasets that contain different characteristics or regions. This way, I could study and compare distinct human or cultural behavior patterns, and the geographic impact on them.
Bibliography


