Pedestrian route prediction from GPS logs

A Thesis submitted for the degree of Doctor of Philosophy

Gavin Smith

Bachelor of Information Technology (Honours) (Advanced Computer and Information Science), University of South Australia

School of Computer and Information Science

University of South Australia

January 2012

Contents

List of Figures

List of Tables

Table of Abbreviations

CRAWDAD	Community Resource for Archiving Wireless Data At Dartmouth
CTW	Context-Tree Weighting
DBSCAN	Density-Based Spatial Clustering of Applications with Noise
EM	Expectation Maximization
EMD	Earth Movers Distance
EPSRC	Engineering and Physical Sciences Research Council
GLMM	Generalized Linear Mixed Model
GPS	Global Positioning System
HDC	Hausdorrf Distance Criteria
HMM	Hidden Markov Model
ICA	Independent Component Analysis
KD-Trees	k-Dimensional Tree
LZ	Denotes compression algorithms by Abraham Lempel and Jacob Ziv
M-Tree	Metric Tree
MAE	Mean Absolute Error
MDP	Markov Decision Process
NP-Hard	Non-deterministic Polynomial-time Hard
OSGB	Ordnance Survey of Great Britain
PCA	Principal Component Analysis
PMF	Probability Mass Function
PPM	Prediction by Partial Match
PQM	Probabilistic Quality Measure
PST	Probabilistic Suffix Tree
RLD	Relative Linkage Disequilibrium
RMSE	Root of the Mean Square Error
ROC	Receiver Operator Curves
SPM	Sampled Pattern Matching
TMC	Total minimum Distance Criteria
TREC	Text REtrieval Conference
WLAN	Wireless Local Area Network

Common mathematical symbols

This list is provided for clarity, with all symbols more fully introduced in-text. All other symbols should be considered to be only locally defined as per the in-text declarations. Angle brackets denote ordered sets and curly braces denote unordered sets. In this thesis location is encoded both as a continuous quantity in a two-dimensional plane and as a set of symbols representing a partitioning of a fixed spatial area. This is distinguished consistently though a slight notation change for clarity. When a symbol refers to the latter encoding this is denoted by using a sans-serif font. This is shown explicitly in this table.

Table 1: **Common mathematical symbols**

Symbol definition	Description
$D = \{H_1, \ldots, H_{\lvert D \rvert}\}$	A set of historic trails (the unordered data set of historic trail observations)
$H = \begin{cases} \langle h_1, \ldots, h_{\lvert H \rvert} \rangle & or \\ \langle \mathsf{h}_1, \ldots, \mathsf{h}_{\lvert H \rvert} \rangle \end{cases}$	A historic trail observation (an ordered set of point observations)
h	A point observation (feature vector) including continuous location coordinates. E.g. $h = [x, y]$
h	A point observation (feature vector) where the continuous location has been mapped to a symbol $s \in S$ where S is a finite set of symbols and each s represents a non-overlapping spatial region. E.g. $\mathsf{h} = [s]$
$\mathcal{K} = \{K_1, \ldots, K_{\lvert \mathcal{K} \rvert}\}$ *where* $K \in D$	A subset of the total historic trail observations used to train a prediction algorithm.
$\mathcal{T} = \{T_1, \ldots, T_{\lvert \mathcal{T} \rvert}\}$ *where* $T \in D$	A subset of the total historic trail observations used to test a prediction algorithm.

Table 2: **Common mathematical symbols continued...**

Symbol definition	Description
$\mathcal{P}(\mathcal{K}, \cdot)$	A predictor trained using the observations in $\mathcal{K}$. The $\cdot$ represents an arbitrary input observation. Replacing the $\cdot$ a given E provides a prediction R.
$E = \begin{cases} \langle e_1, \ldots, e_{\lvert E \rvert} \rangle & or \\ \langle \mathsf{e}_1, \ldots, \mathsf{e}_{\lvert E \rvert} \rangle \end{cases}$	An observed ordered set of point observations making a partial trail as seen so far (evidence) used as input to a prediction algorithm. When evaluating predictors E is one part of a historic observation, $T \in \mathcal{T}$ with the subscripts referring to identical indexes within T.
e	a point observation (feature vector) of the same type as h
e	a point observation (feature vector) of the same type as h
$R = \begin{cases} \langle r_{\lvert E \rvert+1}, \ldots, r_{\lvert R \rvert} \rangle & or \\ \langle \mathsf{r}_{\lvert E \rvert+1}, \ldots, \mathsf{r}_{\lvert R \rvert} \rangle \end{cases}$	For a given E and $\mathcal{P}(\mathcal{K}, E)$, R is the prediction result which is an ordered set of predicted point observations.
r	a point observation (feature vector) of the same type as h
r	a point observation (feature vector) of the same type as h
$F = \begin{cases} \langle f_{\lvert E \rvert+1}, \ldots, f_{\lvert T \rvert} \rangle & or \\ \langle \mathsf{f}_{\lvert E \rvert+1}, \ldots, \mathsf{f}_{\lvert T \rvert} \rangle \end{cases}$	A known continuation of an partially observed trail, E. Used in the evaluation of predictors F is the remainder of the trail T from the test set $\mathcal{T}$ from which E was constructed.
f	a point observation (feature vector) of the same type as h
f	a point observation (feature vector) of the same type as h

Abstract

Pedestrian movement prediction and in particular, route prediction, can provide valuable information to location-aware services, enabling, for instance, ahead-of-time notifications specific to the route, data preparation and efficient pre-fetching and data caching. Predicting human movement, however, is a complex task and this thesis extends state-of-the-art in a promising approach, namely prediction based on the construction and application of statistical models and data mining techniques to large volumes of historic mobility traces. Such data is rapidly becoming available in vast quantities, with movement data able to be contributed back by a growing number of location-aware consumer devices such as smart phones. The approach predicts future movement based on patterns in large sets of past movement, based on assumption that human movement is somewhat regular, an assumption which is both intuitive appealing and has been verified in recent studies. This thesis provides a comprehensive body of work on fine-grained pedestrian level route prediction, for which limited prior research exists, with most previous investigations focusing on (1) prediction after quantization to regions of interest, (2) predicting single destinations, either final- or next-step, or (3) focusing on the transportation domain where movement is constrained to a known road network.

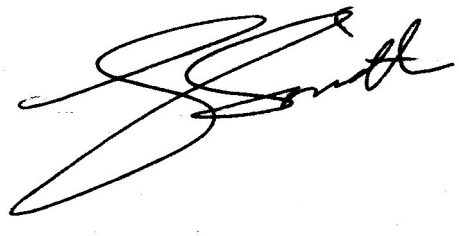
Following an introduction motivating and highlighting the challenges of pedestrian route movement prediction (Chapter 1), the first part of this thesis reviews previously proposed techniques with respect to movement prediction in different situations in the context of pedestrian route prediction from fine-grained logs (Chapter 2). Following this the evaluation of prediction techniques is examined and extended. Noting a lack of consistency in evaluation (for example, only one third of relevant literature share a common evaluation metric) and lack of statistical analysis, a set of current *best practices* are developed providing simple guidelines for researchers wanting to compare movement route prediction algorithms and additionally providing a solid framework for evaluations throughout the thesis (Chapter 3). The state-of-the-art in prediction models for route prediction is then extended with the examination of modelling techniques and the subsequent development and evaluation of novel prediction algorithms. Specifically the reliance on accurate sequence information in historic and input routes in current state-of-the-art predictors is questioned noting that, when considering data aggregated from sources such as consumer grade global positioning systems (GPS), recording errors and other

inaccuracies often violate this assumption. Acknowledging this, and the computational challenges incurred by not assuming accurate sequence information, two novel predictors are proposed in Chapter 4 and Chapter 6, respectively aimed at resource limited and high powered devices. The results show that viable solutions exist to the computational challenges and more accurate predictions can be obtained in this fashion. In conjunction with the proposed models an investigation is made into the use of measures other than conditional probability for measuring the utility of predictions in large data sets where the input matches multiple different historic routes equally well (Chapter 5) concluding that while conditional probability is a good choice it is not the only choice with a number of other measures performing equally as well with potential theoretical benefits. Finally an alternative form of navigational information from the virtual domain, image search click-through data, is investigated via an in-depth user study showing that incidentally, left trails by users within this space are accurate enough to use as a basis for predictions (Chapter 7).

Declaration

I declare that:

- this thesis presents work carried out by myself and does not incorporate without acknowledgment any material previously submitted for a degree or diploma in any university;
- to the best of my knowledge it does not contain any materials previously published or written by another person except where due reference is made in the text; and all substantive contributions by others to the work presented, including jointly authored publications, is clearly acknowledged.



Signed:.....

Date: 17 January 2012

Acknowledgements

Throughout my PhD I've been lucky enough to spend a great deal of time outside of the University of South Australia and Australia in general, meeting and working with a large number of people who are both interesting and generally just good fun. Therefore, in opening my acknowledgements I'd like to extend thanks to everyone I've met in my travels, at home and abroad, overall it has simply been a great three years.

From the University of South Australia, I'd like to firstly thank my primary supervisor, Helen Ashman for giving me the motivation to undertake this PhD in the first place, for providing me with the opportunity and contacts from which the majority of my travel was based and her constant support and advice. I'd also like to thank Ivan Lee, my secondary supervisor, for his support and general discussions. Then of course I'd like to say a big thanks to those I have worked with and along side me in our group and at UniSA in general. Of everyone I would like to explicitly thank Jan-Felix Schmakeit who has helped me out on many occasions and generally been good fun to both work and travel with. Finally I'd like to extend a huge thanks to Chris Brien, who aided in the statistical design and evaluation of my first major study, increasing my motivation that has led me to acquire a much more in-depth statistical understanding, which is reflected in this thesis with a chapter dedicated to evaluation methods. On this note I'd also like to thank Cees Vandereijk from The University of Nottingham who also provided statistical advice while I was visiting.

Throughout my travels I have been fortunate enough to met a number of exceptional people of who I am in now in regular contact and who have played a large hand in the direction my thesis has taken. Of primary note is James Goulding, Tim Brailsford and Mark Truran, all of whom I can not thank enough for their support, collaboration, interest, encouragement, advice and, importantly, ensuring that my time in the UK (and in fact a number of other countries) was a great deal of fun. Additionally I'd like to thank John Miller and the team for having me as an international student volunteer at two World Wide Web conferences, it was a blast.

I'd also like to thank the people I worked with and spent time with at Southampton University and specifically Jonathon Hare and Paul Lewis for having me in their group for three months. Similarly I'd like to thank everyone in the Horizon Digital Economy Institute at The University of Nottingham where I spent six months - a great place to

work due to great people.

My PhD, my travel and the subsequent opportunities would not have been possible without the financial support from the APA Scholarship funded by the Australian government and the scholarship topup and personal development fund provided by the University of South Australia. In addition I'd like to say a huge thanks to the board, contributors and everyone else who continually make the Maurice de Rohan International Scholarship possible, of which I was a recipient. The time abroad it enabled made a significant impact on my PhD for which I am grateful.

Last but not least I'd like to thank my family and my girlfriend, Anne Quandt for their invaluable support.

Chapter 1

Introduction

Movement and location play a large part in our daily lives providing new opportunities for location based applications. In particular, the adoption of global positioning system chips (GPS) within mobile phones provides meaningful locational data at the level of the individual, in contrast to the previous, coarse-grained cell tower or wi-fi location data. This increased precision has, and continues to, enable an increasing range of tailored services, such as Google maps, location-aware search and navigation support systems, assisting our day-to-day tasks and providing novel leisure applications (e.g. foursquare). While understanding the user's immediate locations allows a great deal of applications, understanding and reasoning about possible future locations, and ideally route information, provides a much greater opportunity for personalised services and more generally *intelligent mobile agents*. Understanding future possible locations and routes provides invaluable information to help services to identify what information they should deliver, when, if at all, the information should be presented and importantly what information should be prepared and sent ahead of time to the mobile device due to operating constraints such as connectivity, bandwidth and battery.

Human movement, however, is a complex phenomenon and its prediction non-trivial. Occurring in a complex and ever-changing environment, systems based solely on rules are exceedingly complex to develop if not impossible. As such one of the most promising approaches to predicting human movement has been to take advantage of aggregated human intelligence left by the users of these location-aware devices. In this way users of location-aware services not only utilise the services but also contribute back to it implicitly via their owner's movement decisions. Based on the movement logs kept by these devices, or devices with which they make contact, research into movement prediction via

statistical models utilising this information has been steadily increasing. Many different approaches exist, however, with approaches differing in varying aspects including what type of location data they use, the features they extract, the specific structure and hence assumptions they enforce as well as vast differences in the way erroneous data is dealt with. Based on the idea of aggregating and using the encapsulated intelligence provided implicitly by those contributing each individual movement pattern, the approaches can be considered to be utilising the wisdom of the crowds, relying on the fact that human movement is somewhat regular. Such an assumption has solid grounds, with a recent high profile study [77] into understanding human mobility patterns demonstrating that "human trajectories show a high degree of temporal and spatial regularity".

Work in the field initially focused on applications using the coarse-grained location information provided by cell towers to predict mobile user's movements in order to ameliorate coordination of cell service provisioning (e.g. [38, 148]) and later WLAN hand-offs (e.g. [127, 146, 182]). Based on these coarse-grained location information sources and consequently making coarse-grained predictions, researchers were able to show acceptable levels of next-location prediction accuracy with relatively simple models (see, for example, the comparison of techniques in [182]).

More recently a large body of work has developed from the use of logged GPS trails for movement prediction in the field of transportation where satellite navigation devices have been become commonplace. In this field prediction techniques have been motivated by applications such as improving transportation system management and traffic flows (e.g. [102]), ahead-of-time notifications about upcoming traffic hazards or providing well-timed information about upcoming points of interest [69] and improving hybrid vehicle efficiency by incorporating upcoming road conditions [102].

Location-aware personal mobile devices with embedded GPS, such as smartphones, are now common and have additionally fuelled research into movement prediction at the level of the individual. It is in this domain that this thesis is focused. Differing from previous contexts in both scale and complexity, new techniques have been proposed in an expanding field of research which has seen the idea of the *intelligent mobile agents* further explored. Examples have included locative reminder systems [132], location based advertising [109] and intelligent navigation aids to alert people with cognitive disabilities when they had deviated from their normal route [150].

Current research has only touched on the possible applications enabled by the availability

of highly accurate movement predictions, with many of the applications having vastly differing properties. For instance the use of coarse-grained cell tower location information is only valid for certain applications where the resulting coarse-grained predictions are acceptable. Inherently different to prediction from fine-grained prediction data, the data is characterised by shorter contexts with next step predictions relatively more important. Additionally prediction of vehicle routes has many different properties to predicting pedestrian routes, with the road systems presenting a much more constrained environment allowing the precision of the incoming GPS readings to the corrected using techniques such as map matching. Additionally the road networks help to limit the possible prediction routes. Finally applications that desire route predictions place extra demands on the prediction system preventing the high level quantization of data into logical locations which simplifies the problem in terms of both computational complexity additionally abstracting away issues of noise inherent in the use of consumer grade sensors. For instance [137] investigate prediction by first identifying regions of interest such as train stations and museums.

1.1 Problem definition: route prediction

With the exception of chapter 7, which considers an alternative data source for which route prediction is of interest, this thesis primarily addresses the problem of route prediction from fine-grained location data. In this thesis it is assumed that each individuals' location data streams are able to be segmented into logical *trails*. Trails refer to sequences of point observations that start and end at known or inferred locations. New methods for the inference of such locations is beyond the scope of this thesis and hence segmentation is not specifically addressed here. However, many techniques exist including the use of loss of signal [132], pauses in movement [69] or other more involved learning methods [119]. These existing techniques are discussed more fully in chapter 2 for completeness as part of section 2.2 on encoding GPS data.

Trails rather than per person location sequences are used as they enable the encoding of the definition of a route as a pre-processing step. This allows the thesis to focus on the prediction algorithms by employing a consistent definition of a route and the same pre-processing step.

Specifically the problem addressed in this thesis is:

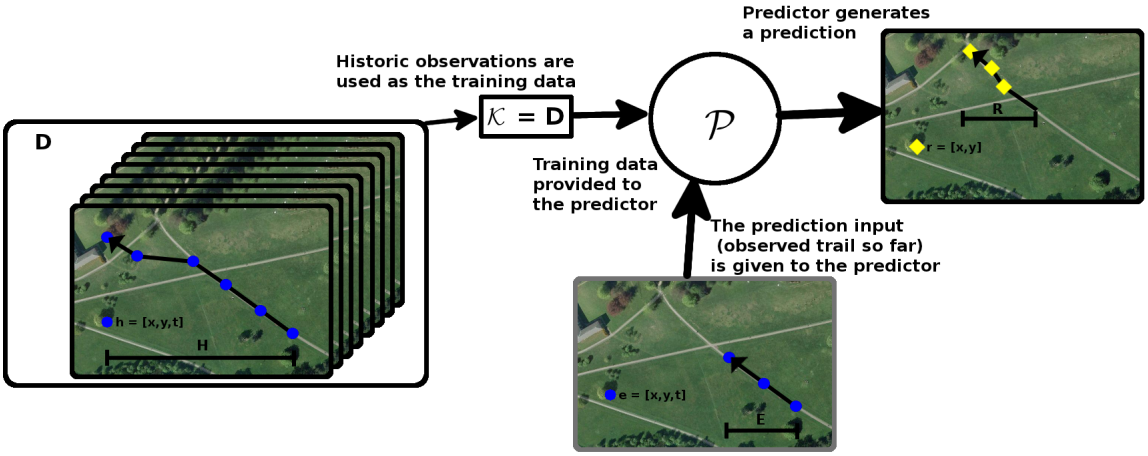


Figure 1.1: Figure visually showing the prediction process labelled with the corresponding symbols used throughout this thesis.

- Given a set of historic trails: $D = \{H_1, \ldots, H_{|D|}\}$ where a trail $H = \langle h_1, \ldots, h_{|H|} \rangle$ and each point is equal to some feature vector (minimally encoding location), for instance: $h = [x, y]$
- And given an observed input trail $E = \langle e_1, \ldots, e_{|E|} \rangle$
- Provide a resulting prediction $R = \langle r_{|E|+1}, \ldots, r_{|R|} \rangle, |R| \geq 1$ of arbitrary length indicating the route that is expected to be taken, where r is a feature vector encoding only locational information (e.g. $r = [x, y]$).

The historic trails are typically used by the prediction algorithms as training data, $\mathcal{K}$. In real world use all historic data would be used as training data, so $\mathcal{K} = D$. However, as will be discussed in chapter 3, for evaluation purposes this is not the case and so throughout this thesis the separate symbol $\mathcal{K}$ is used to denote the training data set.

Figure 1.1 visually illustrates this.

Note that H, E and R are all ordered sets. The definition of route here is made explicit by the definition of r_k as a feature vector of locational data (e.g. $[x, y]$ or $[longitude, latitude]$) only. The exclusion of other features in the output, and hence the evaluation[1] of route prediction algorithms, is intentionally made to focus on route prediction and not on predicting the general context[2] or on predicting explicitly timed

[1] see chapter 3

[2] Where the context is defined as the information encoded in the feature vector p_j.

routes. Such a decision is taken so as to not conflate the accuracy of algorithms with respect to the prediction of a route with the prediction of the timings of the route. This is important due to the relevance of route prediction without timing information. For instance routes are generally considered to be continuous travel by the individual[3], may be quite short and timing information can be generated post-hoc if required (e.g. based on the pedestrians current speed). An example of an application where timing information is less important is location-based advertising, where predicting accurately that pedestrian will go past the potential advertisers shop is more important than the exact time that person will go past, especially considering you know they will go past soon.

Finally it is of note that in the definition of route prediction the input context is not restricted. While in general it is likely that by including additional contextual information an increase in prediction performance could be obtained, in this thesis only the encoding of time and locational information is used. This focus aims to increase the performance of the prediction algorithms under the minimal amount of information expected. It is of note that in the case of the novel predictor proposed in chapter 6 the extension to multiple contexts, both in the input and output, is trivial. Extensions of the other algorithms are left as future work.

1.2 Challenges of pedestrian route prediction

This thesis primarily focuses on the use of GPS data as the movement data from which predictive systems are developed, deviating only in chapter 7 where a different source of movement data is evaluated with respect to the first challenge discussed below, the challenge of mining incidentally-collected data. Despite this focus it is worth highlighting that many of the challenges, such as noise and feature encoding, apply more broadly to forms of data other than GPS location sequences. Specifically, only generic locational data is used as input to the proposed and evaluated algorithms, enabling the direct application of the algorithms to different locational data. Therefore, while the majority of the discussion takes place in the context of GPS data, the approaches can be applied more generally.

[3]This is generally enforced by the way the trails are created from the raw GPS streams in the first instance with a common approach defining trail start and end points based on points of stoppage.

Challenges with respect to modelling and predicting pedestrian movement from GPS data can be roughly grouped into five different categories. Firstly there are the general challenges associated with mining incidentally-collected data. These include challenges relating to noise within the data and varying sample rates. Secondly there is the challenge of selecting what data to record and process (feature selection) and the tighly coupled challenge of how to encode the selected features (feature encoding). For example, in the case of location data, *should location information be on a continuous scale? Should a set of logical locations be extracted? Or should a basic grid quantization be used?*. The third challenge is the development of the predictive algorithms themselves. This includes how they internally represent (model) the historic data and the algorithmic mechanisms that allow predictions to be made both accurately and in a timely fashion. The fourth challenge is that of evaluation. In other words, given a prediction algorithm, *how should it be evaluated?* While all previous research has addressed this, no real consensus has been reached in the context of route prediction. Finally the fifth challenge surrounds data collection, with the primary concern being privacy.

1.2.1 Challenges of incidentally-collected data

Incidentally-collected data has a set of unique challenges in many cases. This is especially true in the case of GPS data which is the data type focused on in the majority of this thesis, but also in other cases, as highlighted by the examination of logs indicating movement among search results on the World Wide Web in chapter 7. In both these cases the incidentally-collected data is prone to significant and uncontrolled levels of noise, since the data collection mechanisms were not developed specifically with the application of prediction algorithms in mind.

With respect to GPS, consumer grade embedded units typically found in most mobile devices are of varying quality and therefore produce location readings of varying accuracy. GPS accuracy is also sensitive to a large range of factors including satellite positions, signal obstructions (such as buildings), atmospheric conditions and even where the device is held/placed on the user [89, 189]. In general it is also expected that a wide range of different devices will contribute to the data set. This only exacerbates the previous issues, adding complexities associated with varying sample rates and the possibility of various *error correction* mechanisms built into specific devices such as automatic dead reckoning when the signal is lost (as highlighted in [11] for example). While it is pos-

sible to construct discrete prediction systems for individual devices, such a practice is suboptimal since the statistical prediction models generally perform better with more data, and always perform poorly with very little. Considering the vast array of consumer devices, the latter may very well occur in a real system. In summary, since the data must be assumed to be heterogeneous in device type and generally of relatively poor quality, very minimal guarantees with respect to correctness, sampling rate or in device error handling can be made. Ways of addressing these issues are discussed in chapter 2, section 2.2.

With respect to movement data among search results, as considered in chapter 7, the challenges of incidentally-collected data are generally characterized by errors not in the recording of the clicked resource, but in the persons unintended movement with respect to their current task. Such false navigational steps are much less common in spatial movement where the user can typically see (at least in the short term) and/or have a good idea where they are going and where the physical effort of false steps is much higher. In navigation among search results the assumption that people will not make false steps is tenuous [99], with people often making decisions out of curiosity and/or with very little information about the subsequent page. Since the utility of many prediction applications is dependant on making useful predictions with respect to a goal (i.e. an information/destination goal) noise of this kind can be particularly detrimental. Therefore, the severity of these issues and the subsequent potential use of this form of data for movement prediction is discussed in Chapter 7. This step is an important pre-cursor to the application of prediction algorithms similar to those presented in this thesis. As noted in Chapter 8, however, such application is left as future work.

1.2.2 Challenges in feature selection and encodings

Prediction systems can only reason about features they know. These features are generally either extracted from the real world either via sensors or provided as input. As such a major challenge is identifying the features from which the models reason, more succinctly the question, *what factors need to be modelled in order to make good predictions?*. For instance, should only location (e.g. latitude and longitude coordinates) be modelled? What about sequence? What about weather? What about personal attributes like age? While all are interesting questions this thesis focuses on the issue of modelling sequence (and consequently to a lesser extent time) since it has typically been embedded in the

structure of previous prediction algorithms making its investigation more complex than altering feature vectors in existing prediction systems. Entwined with feature selection is the issue of how each individual feature is encoded.

A common choice of encoding location, for instance, is to determine significant locations via clustering or other means (e.g. [11, 137]). This helps not only to reduce issues of noise with respect to the raw GPS readings, but also helps reduce the computational cost of the prediction algorithms by reducing the number of distinct items given as input. Such clustering, however, does not meet the requirement to provide route prediction as prediction algorithms based on this data can only predict these locations and not the routes in between. As such other options must be considered and computational issues dealt with via other means. In this thesis two specific instances of feature encoding are discussed. The first relates to encoding location with respect to determining equality, specifically in the presence of noise. This is discussed primarily in chapter 2, section 2.2.1. The second instance is in conjunction with feature selection when considering the use of sequence as a feature. This is primarily addressed in chapter 6 although as feature selection and encoding is part of all prediction algorithms it is discussed throughout the thesis as required.

1.2.3 Challenges in developing pedestrian route prediction algorithms

This is the core challenge of this thesis, incorporating the challenges incidentally-collected data and feature selection and encoding. In addition to these aforementioned challenges, the development of movement prediction algorithms has non-trivial challenges relating specifically to the internal model that each algorithm uses to make a prediction. In this thesis data driven models of various forms are considered and hence this challenge relates to the model the algorithm builds based on the historic data given as a training set. The challenges in this area relate to how much historical context can be considered by the algorithm and how such context is used to ultimately improve the quality of the predictions. An often conflicting but equally important challenge is the challenge of ensuring that the algorithm is tractable.

In comparison to previous work using coarse-grain movement data, fined-grain route prediction involves larger contexts. In addition while much previous work has utilized

a limited number of discrete locations, such techniques are not applicable and either the continuous nature of the data or vast numbers of discrete locations must be considered.

1.2.4 Evaluation challenges

Evaluation is clearly an important task. However, with respect to movement prediction, numerous different evaluation metrics have been used, with many not being able to generalise to the output from other types of prediction (e.g. next-step versus route prediction). Clearly this is far from perfect, preventing even approximate comparisons of work within the domain. In chapter 3 a number of evaluation metrics are examined and a solution proposed for the class of movement prediction algorithms that provide some notion of sequence. In addition the challenges with respect to systematic data reuse in evaluation, as is generally required due to relatively small data sets, is discussed noting that normal statistical procedures can not be used since systematic data reuse results in dependent samples. To this end a set of current *best practice* procedures for the statistical analysis of movement prediction algorithms is outlined in chapter 3.

1.2.5 Data acquisition challenges

There are many data acquisition challenges, some of which are subject to much debate. In this thesis, the challenges of data acquisition are not addressed, with the exploration of these issues significantly more involved than could adequately be covered within the bounds of this thesis. Of primary concern is the often-raised problem of privacy and the challenges and ethical considerations that must be taken into account when collecting such data. While in most cases this data is already collected either by explicit opt-in schemes (such as Foursquare[4]) or via the terms and conditions of the use of other location services (such as the location services part of Apple products[5]) much discussion has arisen, with examples of concerns such as "Revealing the location of your home to people you do not want to give your address to" and "Being stalked" reported in user studies in [187]. In response many privacy preserving schemes have been proposed (e.g. see [90] with respect to GPS traces). For a collection of work in the area see, for example, [18]. In this thesis the dataset used was the *D-SCENT* simulation dataset

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

[5] `http://www.apple.com/privacy/` accessed 16th March 2011, section: *Location-Based Services*

[168]. The D-SCENT dataset was generated from an augmented reality simulation that was developed as part of the D-SCENT project funded by EPSRC at The University of Nottingham in the UK. Participants took on the role of workers constructing an Olympic site, performing a host of purchasing and building tasks. The simulation area was fixed and featured 12 locations covering a $80,000m^2$ spatial area. Sixty participants interacted with the game via G1 smart phones and their GPS data was collected every 5 seconds by a central server, with a log only being kept if the participant had recorded movement of at least 5 meters. The resulting data set contains over 30,000 GPS position readings. Therefore in the creation of this data set many of the normal privacy issues were avoided, with participants only being recorded on supplied devices within the fixed area during the game to which they had consented to play. It is important to note that while tasks undertaken by participants were artificial, their movement across the simulation area was completely unconstrained and reflects dense, real spatial behaviour. However, for research to adequately progress real world data sets are required, and must be open, at least to the research community. In this case the issues of privacy have to be fully taken into account. This has already begun with repositories such as CRAWDAD[6], although currently only one data set [162] relates to GPS logs from pedestrian level movement. Compared to the D-SCENT data it is significantly smaller in terms of participants (between 8 - 32) and over a larger area meaning that in the overall data set the traces significantly more spread out providing less chance for the algorithms to learn compared to the D-SCENT data and what would be expected in reality.

1.3 Overview and list of contributions

This thesis extends state-of-the-art in movement prediction from GPS data, explicitly focusing on low level route prediction as opposed to either next step, next goal, destination or region of interest predictions. While considered in the case of vehicle movement, low level route prediction has not seen comprehensive investigation in the case of pedestrian prediction for which GPS readings tend to be less accurate, cannot be *snapped* back to a known movement grid and has no restrictions on the types of movement patterns.

Throughout this thesis the core challenge of the development and examination of prediction algorithms is addressed and evaluated empirically though the comparison of the

[6]http://crawdad.cs.dartmouth.edu/

proposed algorithms to baseline algorithms from the literature (chapters 4 - 6). Ensuring a correct comparison and in light of a lack of a set of standards for evaluating movement prediction algorithms chapter 3 develops a framework for such empirical comparisons.

Informing the development of these successful algorithms a critical review of previous work in the field is presented in chapter 2. This chapter considers the core challenge of developing pedestrian route prediction algorithms in section 2.1. Additionally the challenge of using incidentally collected data is considered through the examination of noise within movement datasets in section 2.2. This is achieved though a review of data sets reported in the literature and through the examination of the data set used throughout this thesis. The success of the subsequently developed prediction algorithms attest to the utility of this chapter.

Finally the thesis once more considers the challenge of using incidentally collected data, with final chapter considering the utility of another form of movement data where the proposed algorithms have potential application, providing a clear conclusion based on a user study involving 67 participants.

The chapter-by-chapter contributions of this thesis are detailed below:

- Chapter 2 provides an in-depth critical discussion of previous methods in movement prediction, examining their applicability to the specific case of pedestrian route prediction for which minimal past literature exists. Additionally the properties of noise specific to GPS data is examined in-depth both with respect to observations from past literature and observations from the new data set utilised in this thesis.

- Chapter 3 addresses the absence of a set of standard practices in evaluation within the field of movement prediction from logs of low-level behaviour. This is highlighted by the fact that within papers in the field only a third share a common evaluation metric with another paper or compare results to other techniques and none report any levels of statistical significance associated with their findings. This is addressed by motivating and contributing a set of current *best practice* procedures for the task of statistical analysis of movement prediction results, including the recommendation of an alternative distance metric which provides a more accurate measure of the distance between two routes.

- Chapter 4 contributes a computationally efficient algorithm for prediction pedes-

trians routes from GPS data, motivated by the benefits of utilising movement prediction in mobile devices with limited resources. While computationally efficient, the algorithms shows performance close to that of more complex, full order predictions, in experiments utilising 5m grid based quantization.

- Chapter 5 contributes an investigation into the use of measures other than conditional probability for measuring the utility of predictions in large data sets where the input matches multiple different historic routes equally well. The investigation is driven by the theoretical arguments put forward in numerous papers from the data mining community (see, for example [71]). In addition a new form of prediction output, probability density functions (heatmaps) is proposed and a hybrid approach to combining the spatial matching score (between the input and historic routes) and the measure of utility of the prediction is examined.

- Chapter 6 makes two significant contributions. The first is with respect to modelling challenges in the form of investigations into the relaxed modelling of time/sequence by treating it as part of the feature vector from which it can be modelled arbitrarily. The second is in the form of a novel prediction algorithm allowing efficient computation of such models using multiple iterators over an augmented version of the cover tree proposed in [21]. Results show that certain time-relaxed encodings can outperform current state-of-the-art predictors.

- Chapter 7 looks toward alternative sources of reliable, but incidentally collected and logged navigation data contributing an in-depth user study involving over 67 participants. Specifically the chapter examines the reliability of click-through data from web based image search as informed movement in the informational space provided by the search engine. The study is provided in light of prior studies into document search based click-through data showing that the movement (clicks) is poorly informed with only 58% of the recorded clicks moving to pages considered relevant. In contrast, in this contribution, the alternative form of click-through data from image search is investigated showing a notably higher level of reliability, with clicks being reliable on average 84% of the time.

It is of note that chapter 7 was performed at an earlier stage in the PhD candidacy. The chapter details and evaluates the utility of another form of navigational data that can be collected implicitly, this time on the World Wide Web. While showing promise a large enough data set was not available during the candidacy and therefore GPS data was

used, leading to the contributions outlined above in the other chapters. Publications relating to work based on chapter 7 are listed below:

1. G. Smith and H. Ashman. Evaluating implicit judgements from image search interactions. In *Proceedings of the Web Science Conference: Society On-Line*, Athens, 2009.

2. G. Smith, T. Brailsford, C. Donner, M. Truran, J. Goulding and H. Ashman, Disambiguation from Web search selections. In *Proceedings of the 2009 workshop on Web Search Click Data*, Barcelona, 2009.

3. H. Ashman, S. Chaprasit, G. Smith and M. Truran. Implicit association via crowd-sourced coselection. To appear, *ACM Conference on Hypertext and Hypermedia*, Eindhoven, 2011.

Publications for the work in other chapters is pending.

Chapter 2

Movement prediction in the context of GPS data

Human movement in general is a complex phenomenon with the choice of route and destination dependent on an array of complex factors. Despite this, the benefits of movement prediction and recently their demonstrated degree of high temporal and spatial regularity [77] has prompted a wide range of prediction techniques. These techniques almost exclusively construct (learn) an internal model from a set of training data provided a head of time from which the prediction is made. With respect to the pedestrian route prediction problem defined in section 1.1 this training data is the set of historic trails, D. Note that while the term *model* can be used in a variety of contexts, throughout this thesis the term refers to these internal models built by the different prediction algorithms/approaches.

Specifically section 2.1 provides a critical review and overview of related work in order to lay the foundations and aid in addressing the core challenge of this thesis, the developing pedestrian route prediction algorithms. This critical review of previous prediction approaches is then followed by a look at the specific problem domain addressed in the majority of this thesis, GPS data, in section 2.2. Specifically this section focuses first on the need and consequence of encoding such data using techniques employed in prior work and secondly on the process of trail identification. Both are generally preprocessing steps performed on the data before it is used by the predictive techniques with the characteristics, quality and form of the data effecting the utility of the various prediction approaches. Therefore, this section directly considers the two challenges of using and encoding incidentally-collected data identified in chapter 1. In addition, in

this section, the characteristics of the data set used in this thesis is considered as an empirical example of the properties expected from real world data sets for pedestrian route prediction.

2.1 Movement Prediction: A review

Prior work in the broad domain of movement prediction has generally been in contexts in which either the data is able (or desired) to be quantized. Examples include prediction of destinations and the prediction of movement of mobile phones between cell towers. Alternatively a large body of work exists with respect to the prediction of vehicle movements where erroneous location readings can be corrected to some degree by making use of known road maps and assuming the vehicles will only travel on roads. In contrast this thesis considers movement prediction from GPS data, explicitly focusing on low level route prediction for which limited previous work exists. As such, while a large body of literature exists, it is not always clear which approaches are advisable in this specific context. It is this issue that is addressed in this section, providing a strong theoretical basis for the development of movement prediction algorithms subsequently presented in this thesis.

The discussion is started with an introduction into the most basic movement models which formed the initial proposals in the field (subsection 2.1.1). These models are part of a larger category, or modelling framework, called Dynamic Bayesian Networks. Bayesian networks provide a general framework in which movement can be modelled based on observed data.

A Bayesian network is a probabilistic graphical model represented as a directed acyclic graph. Nodes in the graph represent variables, called *random variables*. Intuitively[1] a random variable represents a probability function over a state space. For instance consider a spatial region S quantized into a set of locations via a three-by-three square grid such that $S = s_1, .., s_9$. A random variable, *location*, is a probability function over S indicating the probability of each state $(s_1, .., s_9)$. Thinking of it another way, L can be considered a variable which takes the value s_i with the probability defined by a probability function over S. Edges in the acyclic graph correspond to conditional dependencies between these random variables. In the basic case these random variables

[1]For a formal explanation see, for example, [144].

(e.g. location) is a single, fixed, probability distribution. In the context of movement prediction, however, it is additionally desired to reason with respect to time.

Bayesian networks have been extended to enable reasoning with respect to time. Known as *Dynamic Bayesian Networks* random variables are constructed that represent a probability function over a state space (e.g. location) for a fixed, typically relative, time slice (e.g. now, or the time before now). This results in a distinct random variable per time slice for a given state of interest (e.g. location). In this chapter such variables are denoted using the same symbol but with a superscript representing the time slice (e.g. S^t, S^{t-1} with t representing a time index). The dependency of time is then modelled via edges in the graph. A simple example is shown in figure 2.1 and a more complex example shown later in figure 2.4. This large class of models adds an assumption to the general framework, namely that an event can cause another event in the future, but not vice-versa [73]. It is important to note that this framework provides the ability to develop a wide variety of models. These models can which range from very general and almost entirely data-driven (under some basic assumptions), to well-specified models which utilize and rely on external knowledge of the world provided by the model designer. In all cases, however, the problem space is defined in the structure of the model and as such these models can be thought of as partially data-driven models, where the parameters are data-driven but the model structure is not. More fully data-driven models have been proposed. The most-heavily used can be though of as an extension to simple Bayesian Network models (Markov models) allowing variable length contexts. These approaches make up part of a class of algorithms known as Universal Predictors.

The discussion is continued in section 2.1.2 the more complex Dynamic Bayesian Network models are discussed under the category of *partially driven data models*. Following this section 2.1.3 discusses more fully data driven models including Universal Predictors and an alternative paradigm for movement prediction methods, pattern matching. In contrast to Bayesian Network models and Universal Predictors, the pattern matching paradigm provides an alternative framework for addressing movement prediction with many parallels and shared data structures to a number of implementations of universal predictors. Conceptually simple, and allowing the application of domain knowledge to address issues of noise, the pattern matching paradigm has been applied in a number of recent state-of-the-art proposals to modelling movement.

Throughout this section a distinction is made between providing an overview and critical

review of the related work with respect to the task of pedestrian prediction. Specifically, for each class of approaches, the techniques are first briefly described providing the reader with a general overview. Following this a summary section provides a critical review of the approaches with respect to the core challenge addressed in this thesis, the development of pedestrian route prediction algorithms. This allows readers who are either familiar or not interested in the specifics of each approach to easily access the core discussion of this chapter.

2.1.1 An introduction to movement prediction models

The most straightforward movement models fit within the framework of Dynamic Bayesian Networks and maintain the probability of the next state (for example, location) conditioned on a fixed history length. Computationally cheap and straightforward to implement, the models can perform well but only if the process being modelled approximately meets the strict assumptions made with respect to the future's dependence on the past. Importantly, these models provide the conceptual foundation for the discussion of the more complex models.

First order Markov Models

The most basic Dynamic Bayesian Network is a first order Markov model. A first order Markov model contains only one random variable (a global state of the system) and adheres to the Markov property that the next state only depends on the previous state. In the context of movement prediction the random variable typically (at least) represents the location and so a first order Markov model builds a model to answer the question *what is the probability of being in a spatial region next, given the current region?*. Figure 2.1 shows the graphical representation of the process. Recall that in this chapter random variables over the same state space (e.g. the set of all possible discrete locations) but at different time points are denoted using the same symbol but with a superscript representing the time slice with t representing a time index.

This basic form of a dynamic Bayesian network was one of the first techniques proposed for movement prediction from GPS log data [11], although such techniques had been considered earlier with respect modelling movement at the coarse-grain level of mobile phone tower traces to aid in service provisioning [22]. The model, however, has

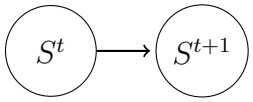


Figure 2.1: First order Markov model. The conditional probability $P(s^{t+1}|s^t)$ is modelled. t represents a time index.

a number of shortcomings. Of primary importance is their lack of predictive power [11]. This is due to the model's inability to model dependencies beyond the last time step directly. By not modelling the dependencies the model loses the ability to discriminate between different inputs where the difference occurs beyond the modelling horizon. Clearly such information loss in general is undesirable and has a detrimental effect on performance.

Markov Models of fixed order

The most basic extension of first order Markov models takes into account fixed histories of a pre-specified length, k, thereby modelling higher order dependencies of order k. Called k^{th} order Markov models, the state of the model depends on the previous k time steps. An example a 3^{rd} order Markov model is shown in figure 2.2. In the case of basic movement prediction where only location is considered these models answer the question *what is the probability of being in a spatial region next given the current and* $k-1$ *previous regions?*

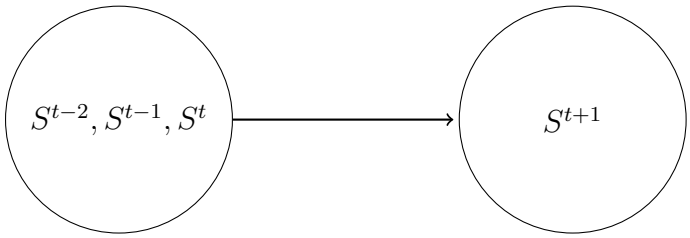


Figure 2.2: Markov model of order 3. The conditional probability $P(S^{t+1}|S^t, S^{t-1}, S^{t-2})$ is modelled. t represents a time index.

Higher order Markov Models were investigated by [11, 22] and later [182] with respect to movement prediction. [22, pg 8] note that "lower order models mislead the algorithm designer by projecting an under-estimate of uncertainty" but show that there will be a limit whereby the improvement gained by increasing the order will be insignificant. This limit intuitively depends on the longest length in the historic data. The authors then go on to highlight that the order required is in fact the length of the input for a given

prediction. The LZ78 compression algorithm is then proposed to automatically arrive at this order for each prediction input. Models based on LZ78 and variable lengths in general are no longer defined by the Markov property, and more over are fully data-driven models, in that the data is now being used to build the structure of the model. These types of models are further discussed in section 2.1.3.

The use of fixed or variable length histories does come with a drawback, however, as the model state space is drastically increased (the number of states is equal to $n \times k!$ where n is the number of states and assuming $k < n$). Due to this, it is much more likely that the input will not match a state that has been seen before and a prediction will not be made. A solution to this problem is to *fallback* [182] to a lower order model and try again. Fallback here refers to the process of using a lower order model if a match cannot be found for the input in a higher order model. For example consider a 2^{nd} Markov model where the random variable in question is location in a world with only nine possible locations $s_1, \ldots, s_9$. Consider the following historic trails from which a 2^{nd} order Markov model is built:

$$s_4 \rightarrow s_2 \rightarrow s_3$$

$$s_6 \rightarrow s_5 \rightarrow s_2$$

$$s_6 \rightarrow s_5 \rightarrow s_2$$

$$s_8 \rightarrow s_5 \rightarrow s_9$$

and the input:

$$s_7 \rightarrow s_5$$

A 2^{nd} order Markov model has not seen the transition $s_7 \rightarrow s_5$ and hence will not make a prediction. Knowing that no prediction could be made it is then possible to use a lower order 1^{st} order Markov model and use only the last seen symbol, s_5, in the input. This then results in the prediction of the location s_2 ($P(s_2|s_5) = \frac{2}{3}$ *vs* $P(s_9|s_5) = \frac{1}{3}$).

In the case of fixed length Markov model this would require additional probabilities to be stored, increasing the size of the model. In variable order approaches, as discussed in section 2.1.3, the process involves simply shortening the input and rerunning the algorithm and does not involve any additional storage costs.

Markov Decision Processes

Markov Decision processes (MDPs) are a mathematical framework for modelling decision making, rather than simply being a model. Markov Decision processes are characterized by:

- a set of known states (S), e.g. a set of locations represented by symbols $s_1, \ldots, s_{|S|}$
- a set of known actions (A = set of all actions, A_{s_i} = set of all actions available from state s_i) that can be performed in each state, e.g. move right
- a set of transitions probabilities ($P(s_i|s_j, a)$) mapping the result of taking an action ($a \in A$) from a state (s_i) to another state $s_j \in S$
- a set of rewards ($R(s_i, s_j, a)$) for transitioning from state s_i to s_j via action a.

A Markov Decision process makes a number of assumptions:

1. Stationary preference: Consider two state sequences, generated by executing a set of actions, that share the same start state (e.g. s_0, s_1, s_2 and $s_0, s_1^{'}, s_2^{'}$). In any subsequences obtained by truncating the sequences by a common offset the order preference should be unchanged. With respect to the example, this means that the order preference between the sequences s_1, s_2 and $s_1^{'}, s_2^{'}$ is the same as between s_0, s_1, s_2 and $s_0, s_1^{'}, s_2^{'}$.
2. Markvoian transition model: The probability of transitioning from a state given an action to another state only depends on the current state.

Given a known destination, MDPs provide a framework for optimally determining a plan of actions to perform at each state to reach the known destination. As such, given the set of Markovian transition probabilities which only describe single state transitions (a first order Markov model), MDPs can be seen as a framework to stitch together the optimal solution given the assumptions.

Of relevance to pedestrian movement prediction is the use of MDPs in imitation learning [160] within the robotics literature. Imitation learning involves making observations which are assumed to be generated from another MDP which acts as a blackbox to the learner. Under this assumption an approximation of the reward function can be learnt resulting in an MDP of similar observable behaviour. If the observations were in fact generated by another MDP then the observations should be reproducible. Since pedestrian movement is not generated from MDP only an approximation of the behaviour

can be expected. The degree of the approximation depends on how closely the observed behaviour fits the MDP assumptions. In [205] this technique was employed utilizing a novel learning technique. Learning a MDP from observations the authors aimed to predict the movement of pedestrians to aid in robot movement planning. Given the current state the destination probability over all states was calculated. Using this information and the plan of action (as calculated by the MDP) the probability of predictions of arbitrary length to the determined destination(s) given the current state were determined, hence generating predictions.

In the context of pedestrian route prediction, it is of note that MDPs can still only model the set of behaviours describable under the first order Markov assumptions (state transitions are only dependent on the current state in the model). As a result MDPs can not utilize the full observed history of a partial trail to discriminate between more complex cases of movement behaviour. It is worth noting, however, that since the learnt model in the case of imitation learning is not necessarily the same as that created by applying the Markov property to the data directly and in some cases the MDP may exhibit superior predictive performance. This is because the only goal of a MDP is to mimic the behaviour detailed by the real movement patterns. However, this is not examined in this thesis.

Summary

In summary, fixed length Markov models of any order are suboptimal predictors if the process being modelled does not match the order assumption made. Even if the fixed length, k, is known at least k consecutive observations need to be made before a prediction can be made, thereby limiting the application of the system. Additionally, without specialized data structures and fallback mechanisms similar to those employed in variable length Markov models, higher order Markov models can easily fail to make predictions while requiring a large number of probabilities to be stored. This quickly becomes intractable for large k. Lower order Markov models, however, still provide the computationally cheapest prediction model on which tractable planning and learning algorithms have been developed. An example is Markov Decision Processes which address the issue of predicting multiple steps into the future in an optimal way given some modelling assumptions and a known destination rather than simple recursive calls to the prediction algorithm.

Considering the specific challenges of pedestrian route prediction it is unlikely that a fixed order model will sufficiently model pedestrian movement. The is because people do not make movement decisions based only on their current state, or a foreseeably fixed number of states, but rather according to high level goals. Recognised in the literature, this has led to two major approaches in the development of prediction models with respect to fine-grained movement prediction. The first seeks to build completely data-driven models, assuming that peoples past behaviour is indicative of future behaviour in general. The second seeks to model certain aspects of the world that cause the behaviour. This is done using knowledge supplied *a priori*. Conceptually this may be the individual's goal. Models following this approach have focused on a class of dynamic Bayesian networks that extend Markov models by introducing unobservable states along with the required corresponding model structure and inference algorithms. In contrast purely data-driven approaches have typically been characterized by variable length Markov models and other pattern matching techniques. In both cases there is also the assumption that enough data exists to build and adequately train the models.

2.1.2 Partially data-driven models

Partially data-driven models seek to model certain aspects of the world that cause the observed behaviour using external knowledge supplied *a priori*. At a high level an example is the individuals' goal, and at a lower level includes things such as road segments and transport modes. Modelled as unobservable states, these higher level variables are modelled using a number of techniques of varying complexity. Techniques applied to movement prediction have included Hidden Markov Models using Kalman Filters for inference, hierarchical dynamic Bayesian networks and conditional random fields.

Hidden Markov Model and the Kalman Filter

A regular Markov model assumes the states and state transitions are known and that at any time the current state is known to be one of the model states. In contrast the Hidden Markov Model (HMM) assumes that the states can not be directly observed, and that a set of observable symbols provide evidence that the system is one of the unobservable states. These unobserved states must be supplied by the model designer. Transitions are still between the unobservable states, but these can not be calculated

directly from the data, since the historic data is only related to the observed symbols. Therefore a fully specified HMM requires:

1. a set of unobserved states
2. transition probabilities between the unobserved states
3. a set of observable symbols
4. a probability matrix describing the probability of seeing an observable symbol given the system is actually in a specific unobserved state

An example of a HMM compared to a standard first order Markov model is shown in figure 2.3. In the figure recall that nodes correspond to random variables. In other words, probability distributions over a state space. In the case of X, the unobserved variables, the state space is the user-defined set of unobserved states. In the case of E the observed variables, the state space is the user-defined set of observable symbols.

Figure 2.3: Graphical representation of a first order Markov model (a) vs. a first order Hidden Markov Model (b). E represents observed variables (evidence), X represents unobserved (hidden) variables.

Prediction can then performed using a Kalman filter. Future positions are predicted using the Kalman filter's predict step repeatedly. Kalman filters seek to predict the hidden variables state by using a two-step process [133]. In the first step a prediction is made based on the previous hidden state estimate and the transition model between the hidden states. In a Kalman filter this model is the transition probabilities between the hidden states combined with a normally distributed variable representing the process noise with a mean of zero and a model specific standard deviation. The prediction step represents the selection of the hidden variables next state with the highest probability given no additional knowledge and is shown in equation 2.1, where x^i represents the state of the hidden variable at time i and e^i represents the observed symbol (evidence) at time i.

$$p(x^t|e^{1:t-1}) = \int p(x^t|x^{t-1})p(x^{t-1}|e^{1:t-1})dx_{t-1} \tag{2.1}$$

In the second step a measurement is taken/received and the probabilities adjusted, calculating the probability given the new piece of evidence. Known as the update step, the equation is shown in equation 2.2.

$$p(x^t|e^{1:t}) \propto p(e^t|x^t)p(x^t|e^{1:t-1}) \tag{2.2}$$

[128, 170, 180] all use such an approach to predict movement, although [128, 170] mainly focus on very short prediction horizons before utilizing the update step of the Kalman filter. As such the systems are aimed more at providing location estimation rather than prediction. Of note is [128] in which an extended Kalman filter[2] is used and the prediction step utilized to provide predictions continuously until a measurement is available. The authors note the importance of receiving frequent position measurements noting that otherwise cumulative errors can result in an estimate with poor accuracy. Such an observation can be explained by examining the prediction equation on its own. In the absence of any update, the model reverts to a simple first order Markov model, where the first order Markov model is the hidden states and the transition matrix between these states. In this case the model returns to simply making the locally most likely decision at each timestep, irrespective of longer term goals [180, 205].

Hierarchical dynamic Bayesian networks

The most notable work done in the area of partially data-driven modelling for movement prediction is the series of papers [119–122, 149, 150] from researchers at the University of Washington, with the bulk of the work the PhD thesis of Lin Liao [118]. The work examines the use of both Hierarchical dynamic Bayesian networks and later the conceptually similar Conditional Random Fields[3]. Hierarchical dynamic Bayesian networks extend hidden Markov models by allowing multiple unobserved variables (model aspects such as velocity with a corresponding set of states) to be modelled in a hierarchical fashion with the bottom layer being the observed variable. The framework provides a natural, graphical, way for the modeller to show how the variables, both observed

[2]The extended Kalman filter first linearizes the system using first-order Taylor series expansion. The Kalman filter predict and update are then updated accordingly. See [23] for more details.

[3]Conditional random fields directly models conditional probability distributions rather than the joint probability distributions, for more information see [1].

and unobserved, being modelled interact. Figure 2.4 shows the model used by [118, ch 6, pg 81] to model movement. The model encodes the observed GPS measurements (observable movement patterns) as being dependent on the location, which in turn is dependent on the transportation mode, which in turn is dependent on the person's goal. The model represents a two-slice hierarchical dynamic Bayesian network, making the simplifying assumption that all modelled aspects are only dependent on the previous time slice.

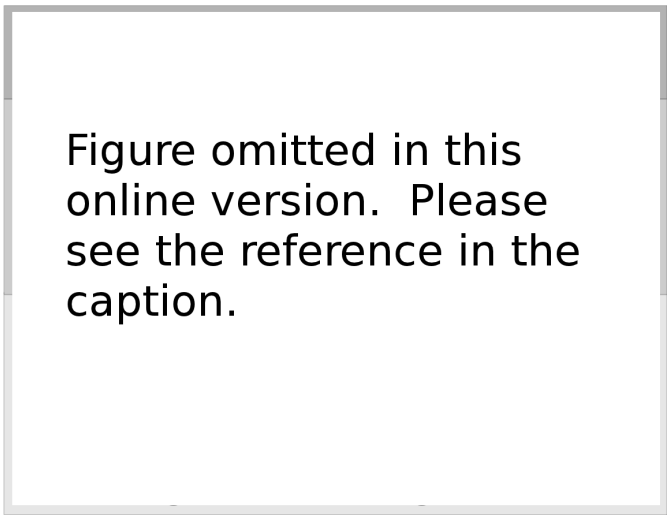


Figure 2.4: Movement model proposed in the PhD Thesis of Lin Liao, reproduced from [118, ch 6, pg 81]

These Bayesian approaches to modelling start with some *a priori* knowledge about the model structure, the previously discussed graph supplied by the model designer including the variable domain (continuous or discrete and the enumeration of states in the case of the latter), and a set of model parameters (probability distributions of certain parts of the model). The model parameters can either remain fixed, or be adjusted so that the models provides a better fit to a set of training data using the Expectation-Maximization (EM) algorithm (for more details on learning Dynamic Bayesian Networks see [73]). In [149] the authors fix all model parameters except the edge transition probabilities (which is conditioned on the previous edge transition and the previous mode of transport) and the probability of mode of transport (which is again conditioned on the previous edge transition and the previous mode of transport). The use of fixed parameters greatly reduces the amount of training data required and reduces training complexity.

Within Hierarchical Dynamic Bayesian Networks (and Conditional Random Fields) inference and learning complexity are of primary concern when designing the models, with arbitrary graph structures being NP-hard and considered intractable in general [83]. As

such simplifying assumptions, such as the first order Markov property used in [122], are required. Additionally approximation algorithms are typically used for inference [118], particularly if variables are allowed to be continuous (as is the case of velocity in [118, 149]). Models utilizing conditional random fields have similar issues and additionally the use of continuous variables have not been explored in the context of movement prediction. This is potentially due to the recent nature of extensions to conditional random fields allowing the use of continuous variables [157] which introduce different challenges to ensuring feasible models [159].

Importantly partially data-driven models have a structure based on the specifically modelled features which makes it possible to reason about these features. For instance in the model presented in [118] the system models the persons goals and can provide alerts if the person's behaviour starts to become closer to a goal other than one specified.

Summary

In summary, partially data-driven models address many of the challenges specific to pedestrian movement prediction. In particular they enable a greater historic context to be learnt by the algorithms from the historic training data. However a number of drawbacks still exist:

1. The model structure is fixed, requiring well-crafted models to be developed by hand using domain-specific knowledge.

2. Learning and inference can be computationally expensive with approximation algorithms required in the case of more complex models

3. The full historic context within the historic trails is potentially underused, by assuming independence beyond the previous timesteps to ensure tractability. While interactions between the hidden variables may completely capture the observed multi-time step dependencies they equally may not. For instance consider a hierarchical model with hidden variables of *goal* and *location* and a set of observed positions with the assumption that the next time step is then dependent only on the past one. This model assumes that given the current goal, the choice at a cross road (between locations) is always only dependent on the current goal and current location. Clearly this is more complex that a first order Markov model over observed positions. The more additional hidden variables or an increase in

the number of states per variable (e.g. the number of goals the person could have relative to the locations) the more complex interactions can occur allowing differing behaviour across multiple time steps.

4. Prediction beyond the short term may quickly degrade in performance, particularly with simple models. Typical inference algorithms such as Kalman and particle filters predict multiple future timesteps by projecting the prediction step. While performing well over short horizons the algorithms can degrade significantly over large prediction horizons due to the lack of information to execute the update step. This is particularly true for Hidden Markov Models where the prediction step is then only using a first order Markov model as the underlying prediction mechanism.

2.1.3 Fully data-driven models

Unlike the partially data-driven models presented in the previous section, fully data-driven models derive their structure from the training data. Specifically the data driven approaches address the prediction problem by learning probabilities for prediction routes for a learnt[4] set of *contexts*. Contexts are continuous subsequences of observations that have appeared in the training data which are then matched against the input. Under the assumption that the use of arbitrary length contexts can provide accurate predictions these algorithms learn both the structure (the contexts) and the probabilities for these contexts.

At this point it is important to make a distinction between the model and the data structure used to implement the model. In basic first order Markov models the data structure is typically just a matrix containing transition probabilities. In fully data-driven models the model is built from patterns contained within the data which is somewhat unknown. Data structures for data-driven models are varied, with the most common being tree-based implementations due to their computation efficiency at matching. Fully data-driven models and variants account for the majority of the proposed approaches within the literature, with proposed approaches varying in model subtleties and in their

[4]Here the term *learnt* is used in a very broad fashion. Note that this may refer to the learning of a set of contexts in advance or to the automatic selection of the best context from the training data at runtime via a fast matching procedure. Perhaps a better, but less succinct, way to describe this would be *derived from the training data*.

approach to dealing with noise. The models and their subtleties are dealt with in this section. Approaches to dealing with noise that are, in general, not specific to any one model are discussed separately in section 2.2. It is of note that fully data-driven models are still considered a modelling problem, with general variable length Markov model algorithms learning the variable structure using the data as a training set while trying to not overfit[5] the model. This is in contrast to pattern matching although, as will be highlighted, in practice they can be very similar.

Universal Prediction & Variable length Markov models

A predictor for a process which is governed by an arbitrary unknown model that performs essentially as well as if the model was known in advance is called a *universal predictor*. Developed initially for compression, Universal Predictors address a slightly different problem than that described in for pedestrian route prediction in section 1.1. Specifically Universal Predictors do not use a set of trails as the training data. Rather a single, long, stream of sequential observations is expected as the training data. Contexts of variable length are then automatically determined from the stream of observations and the probability over all possible states for each selected context is calculated by considering all subsequences of the training data stream. When utilizing Universal Predictors for movement prediction a number of approaches are possible. Considering the GPS data from a single person it is possible not segment the GPS data into trails and use it as a single stream of sequential observations directly. Note that this is not generally desirable since device errors or flat batteries can result in movement occurring but not being recorded. If this movement is not noted and the observations simply concatenated movement between locations that did not occur is artificially introduced. Moreover, however, in general GPS data is acquired from multiple people simultaneously. In this case one could append the GPS streams from each person, however, this would lead to subsequences detailing movement that did not occur at the concatenation points. An example is shown in figure 2.5. The solution, therefore, is to only consider sub-sequences within the individual trails when identifying contexts and calculating the probability over all possible states for each identified context. This approach, for instance, is taken in [31].

[5]Overfitting is the development of a model which has learnt the training data and not the general patterns. In other words the probabilities match the training data, but not do not generalize to the unseen population. A simple example in the case of fitting polynomials to a set of points is fitting a polynomial of equal degree to the number of points.

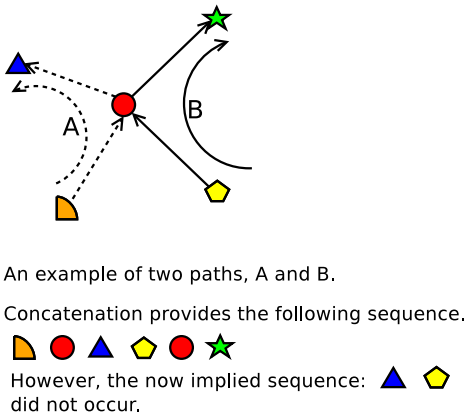


Figure 2.5: An example highlighting an error that can be introduced by not considering individual trails (from a single person or from two separate people) separately.

In the rest of this section it is assumed that such an approach would be implemented and the algorithms discussed considering a single stream of observations.

The general problem considered by the standard by the universal prediction algorithms is:

- Given a sequence of state observations $s_1, s_2, \ldots, s_{t-1}$ where each s_i is a state within a collection possible states S (e.g. the locations that can be occupied)
- Given an input $E = e_0, \ldots, e_k, e_i \in S$ which is provided when requesting the prediction and varies across different predictions with arbitrary length (k)
- Then either, (1) generate the conditional probability distribution $P(S|E)$ over all possible states S, or (2) simply provide a prediction of a state s_t. Since the former is more informative it is generally preferred within the domain of movement prediction enabling predictions to be ranked.

At a generic level universal prediction assumes that the given sequence of observations[6] was generated by either a known or an unknown[7] source distribution. Within movement prediction universal predictors that assume the source distribution is from the family of Markov predictors have been used [52]. The basic assumption, as previously stated, is that the use of arbitrary length contexts (histories) can provide accurate predictions.

[6] in this case the movement sequecence or set of sequences once trails are used and only subsequences within trails are considered

[7] If the source is unknown the data is then considered to be generated in an arbitrary but deterministic fashion.

Universal predictors simultaneous identify a fixed set of contexts and calculate the probability distribution over possible states for the next timestep. Considering all subsequences of the training data algorithms have been developed where the context length is assumed to be either bounded or unbounded. Under each assumption it can be shown that universal prediction algorithms can provide guarantees with respect to achieving known lower bounds on error rates[8] as the number of observations tends to infinity [134]. The case where the maximum length is bounded but determined automatically by the algorithm corresponds to the approach proposed under the name variable length Markov model [30] within the statistics literature, indicating the conceptual extension to the fixed length Markov models discussed previously.

Many universal predictors can be seen to form the basis of many movement prediction methods as well as having been directly applied. Moving beyond the fixed k-order Markov models, universal predictors were among the first to be applied within the movement literature. For instance, [22] utilized a modified version of the compression algorithm LZ78. The application of a number of universal prediction variants followed. The variants differ primarily in two ways. The first is the way in which they handle the *zero frequency problem*, which refers to what probabilities should be assigned to symbols in the alphabet if no match can be made. The second is how the variable length modelling is done. In general this is the problem of how the set of contexts are determined from the continuous (or set of continuous streams when modified for movement prediction) input stream of observations.

LZ Family of compression algorithms: In 1999 [22] examined the use of fixed length Markov models with respect to next step mobile cell tower prediction based on past movement logs between the towers. Noting the issues of fixed length models they proposed the use of a modified version of the LZ78 compression algorithm. LZ78 creates variable length contexts by splitting the training sequence into non-overlapping blocks representing the shortest sequence not seen so far. Each block then forms a branch in a tree based data structure.

The basic approach to identifying the contexts within LZ78, however, has a number of issues for prediction with the inability of the algorithm to correctly maintain frequency counts of subsequences across blocks. This was addressed in [78] using a sliding window

[8] Assuming the source is of a certain class. As noted in the case of movement prediction the assumption is that the source distribution is from the family of Markov predictors.

approach to update all occurrences of subsequences of the current block in the tree. In all cases the maximum depth of the tree and hence maximum context length that can be matched is controlled by the longest, shortest subsequence not seen in the tree up to that point in parsing the input string. While this has proven to work well with data compression, with respect to prediction the approach has shown to be less attractive in general than other universal prediction approaches [14].

Probabilistic Suffix Tree: Like the LZ78 algorithm and variants the basic form of the probabilistic suffix tree (PST) is a tree structure that models the probability distribution for discrete events occurring in a sequence. The approach differs significantly in how it determines the maximum depth of the tree, and in that the tree constructed is a suffix tree. Formally, given a set of states S (for example representing all possible locations) the structure is a tree with:

1. degree equal to the number of states ($|S|$)
2. the nodes labelled by pairs $(q, P(S|q))$ where
    - q represents a sequence of observations $(s_0, \ldots, s_t)$ for which the nodes next step $(t+1)$ probability is conditional on.
    - $|q|$ = the current depth of the tree
3. Child nodes represents an expansion of the historical context q of the current node by a single, earlier observation. Conversely each parent nodes q represents the suffix (one less piece of context) of the current node.

For a given fixed depth k the above structure represents a fixed length Markov model of length k and the size of the tree is exponential in k. Probabilistic suffix trees [164, 165] (also known as prediction suffix trees) alleviate this problem by, given a predetermined maximum depth, pruning the tree keeping only the nodes that have probability distributions *different enough* from their parents. In other words, only when the extra context would make a significant difference to the prediction is the extra context modelled, for some definition of significant. In practice a predetermined maximum depth is utilized to provide a principled notion of two probability distributions being *different enough* [165]. Figure 2.6 shows an example of a PST. As such, PSTs learn contexts which are subsequences of the training sequence which provide distinct conditional probabilities on the symbol in the next time step.

Extensions to PSTs have been proposed which use statistical tests to determine which

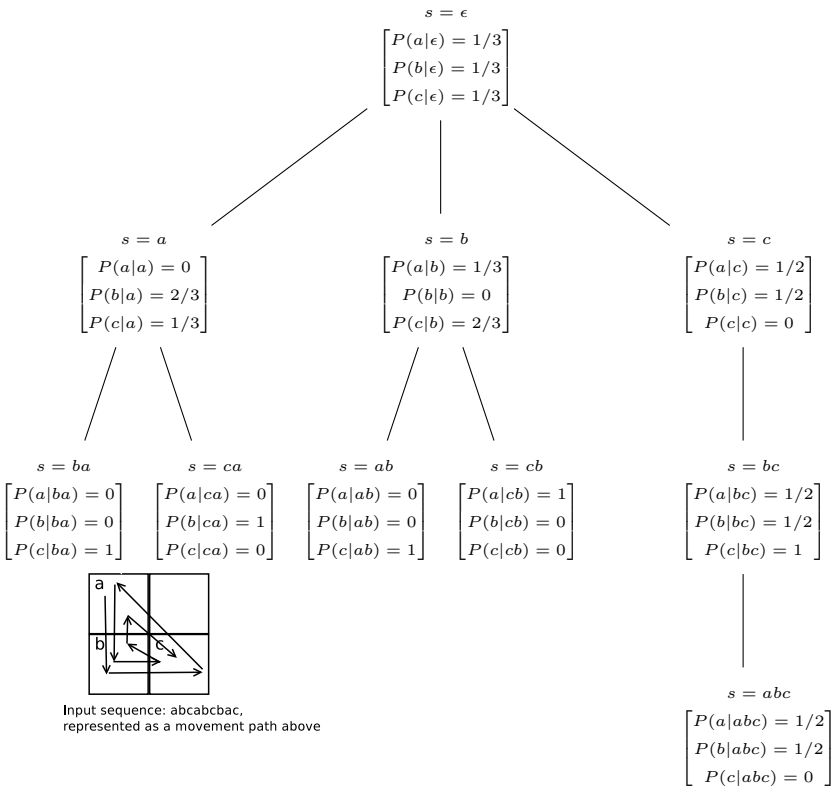


Figure 2.6: Example of a PST with a maximum length of 3 for the sequence *abcabcbac*. Nodes with all zero probabilities are omitted.

nodes should be merged that do not require a predefined bound allowing the bound to grow based on the training data [30]. It is this work which was originally proposed under the name Variable Length Markov Chains in the statistics literature. Additionally work has since been done to consider the case where a bound is not assumed (i.e. it is considered that the data was generated by an unbounded process rather than a bounded process of some automatically determined bound) [53, 66].

Predictions based on PSTs involve matching the input against the tree in reverse order, i.e. starting by following the edge matching the most recent observation in the input. The node reached after matching the whole input then contains the probabilities of the next symbol and the symbol with the highest probability is selected as the next time step prediction.

Within the movement prediction domain [200] proposed the use of PSTs in order to prediction urban vehicle routes. A fixed maximal length K was provided and two thresholds manually set to control when the context should be extended and when it should not, rather than using one of the various automatic methods from [30, 53, 66, 165]. In contrast, while also using a PST structure [147] use domain specific information to create

maximal contexts and the build the tree omitting (pruning) nodes with zero probability only. In the work the authors also forgo the normal storage of probabilities at each node, instead storing a set of metadata and determining the probabilities based on various criteria on the fly. In their work the authors propose the use of standard probabilities based on counts as well as on a notion of time spent in the sequence. With respect to movement route prediction both [147, 200] only seek to predict the next time step. The model and prediction mechanism they utilize (the PST) however, could be recursively called appending the next time step result to the input before the next recursive prediction call.

The work of [147] shows a gradual transition from the universal prediction framework to the general notion of pattern matching. No longer are contexts identified based on the assumption of an underlying variable length Markov model, but rather they pre-specify them using domain knowledge and consequently they can not longer be considered universal prediction algorithms. However, substantial similarities exist as in both cases a variable length Markov model is being constructed in the form of a PST. With the contexts being well defined and no longer just considered sequences of arbitrary data the pruning mechanisms can operate at the level of a trail [139, 147] rather than when adding each node. Extending this further the problem simply becomes one of matching the input to a set of historic trails via a partial matching algorithm [178] and selecting the most likely trail from those matched [137]. Within PSTs the matching is simply exact matching against the built tree although, considering the problem as a pattern matching one, there is clearly a much wider range of techniques available. This transition to pattern matching is further discussed later in this section.

Prediction by partial match (Prefix trees): Prediction by partial match (PPM) is a technique that utilizes variable length contexts to make predictions [45]. Traditional PPM has a similar requirement to PST requiring the maximal order, k to be defined, although an unbounded version has been proposed [44]. A simple way of initially understanding PPM is to consider it to construct a table of probabilities $P(S|C)$ for all the different possible contexts (continuous subsequences) C of length $1 \dots k$ from the training data for all states in S. Predictions are then made by attempting to match a context of equal length to the input. If such a match can not be found the input is reduced by one (the least recent observation in the input) and another match attempted. This represents the *fallback* strategy previously discussed in section 2.1.1. At the lowest level there is assumed to be a set of fixed probabilities for all states conditioned on the

empty set. This simple approach is sufficient for basic sequence, and hence movement, predictions as at some point it will match (or report a fixed probability) and return the most likely prediction. This is sufficient if we are interested in the most likely prediction and not the probability of the prediction. If the probability of the prediction is required then each time a fallback was initiated a probability known as an escape probability is required. The general recursive mechanism of PPMs is, given $\sigma \in S$, $e_i \in S$, then:

$$\hat{P}(\sigma|e_{t:j\ldots n}) = \begin{cases} \hat{P}(\sigma|e_{t:j\ldots n}) & \text{if } e_{t:j\ldots n} \text{ appeared in the training sequence;} \\ \hat{P}(escape|e_{t:j\ldots n}) \times \hat{P}(\sigma|e_{t:(j+1)\ldots n}) & \text{otherwise.} \end{cases} \tag{2.3}$$

Unfortunately the selection of escape probabilities is non-trivial and [45, pg 397] note with respect to the selection of escape probabilities, "...in the absence of any a prior knowledge, there seems to be no theoretical basis for choosing one solution over another.".

Since this work does not focus on the need to determine the actual probability and relative probabilities suffice, the selection of escape probabilities and the various proposals are not further discussed. A common and efficient approach is to implement the PPM as a prefix tree [14]. As such PPMs construct contexts based on all prefixes of each symbol in the training data from of length $1 \ldots k$.

PPM is used within the movement prediction literature in [31] and, while not stated explicitly, in [202]. In both cases prefix trees were used and building the PPM prefix tree is done by considering all sub-sequences within the individual trails. In [31] escape probabilities were utilized since actual probabilities were required as part of the evaluation function used. In contrast in [202] escape probabilities were not used and the accuracy of the resulting prediction from the system compared to a known correct result. Of note is that if the movement data is already split into logical patterns of movement between goals (trails), the fixed length of the PPM can be logically set at the length of the longest trail.

Sampled Pattern Matching: In a similar fashion to PSTs, Sample Pattern Matching (SPM) [92] automatically determines the contexts used that are used to predict a given sequence given an input training sequence. Context is determined in a conceptually straight forward manner by first selecting the longest suffix from the input that is contained in the training data and then taking a fraction (α) of this longest suffix

as the context. As such if $\alpha = 1$ the approach is the same as PPM. For any $\alpha < 1$ the approach trades contextual information for potentially more observed samples from which to construct the next state probabilities thereby helping to prevent overfitting. SPM has not seen much use within recent literature on movement prediction. A possible reason for this is that the problem of overfitting is often dealt with as a preprocessing step (i.e. only frequent movement patterns are mined from a collection and presented to the prediction algorithms [95, 137, 139]). The approach, however, is evaluated with respect to predicting next location given Wi-Fi mobility data in [182] along with a number of different approaches. They conclude by noting no significant difference between the more complex algorithms such as SPM and a second order Markov model with a fallback mechanism. The result, however, is not relevant to pedestrian movement prediction as the data set utilized represented very coarse grain data in which the number of locations a user moves from is expected to be small and quite regular. As such it is not surprising that by using small contexts accurate predictions can be obtained. Within pedestrian movement data long and varied contexts are expected per person and so the results are not applicable in this context.

Context-Tree Weighting: In contrast to other approaches such as PST and PPM, context-tree weighting (CTW) does not seek to determine a set of previous contexts and then match an input to a context in the set. Rather the approach merges exponentially many Markov chains of bounded (user defined maximum) order. Originally proposed for sequences of with binary states only [196], the approach has been extended in various ways to allow states [191]. CTW has shown varied results, named as one of the better universal predictors along with PPM in [14] but shown to perform not as well as PPM in [103]. No studies with real world movement data have been performed. Within the context of pedestrian route prediction it is unclear that merging probabilities from lower order Markov chains is advisable when higher order context is available, since many common corridors of movement exist and it can be the first few movements before entering the common corridor that are the discriminating ones for the prediction.

Concluding remarks: While the use of universal predictors is appealing in theory, the direct application in some practical domains do not always result in accurate predictions. Developed in general for data compression, and then generalized to prediction (since the problems can be shown to be directly related [65]) the general algorithms do not account for noise which introduces additional randomness into the system reducing predictability of the overall data and hence the performance of the system. Addition-

ally the parametrization of the different algorithms can have a large impact on their behaviour for the finite sequences observed in practice, despite not affecting the asymptotic performance [196]. Finally, and most importantly, the application of universal predictors requires the pedestrian prediction problem to be suitably encoded. Even if a perfect universal predictor was realized the encoding of the problem into a set of states, in other words which features to use, is still a notable problem. For instance within movement prediction, consider the problem of a fork in a path, one which leads to an undercover route, the other outside. Assume that when it rains I always take the undercover route, and when it doesn't I take the other and it rains 30% of the time. If only the locations are modelled then the optimal predictor would model my choice as of taking the undercover path as 30% each time. However, if the weather was also encoded the optimal predictor would be able to predict my choice accurately every time.

Context pattern matching

In contrast to Universal Predictors, which were originally developed assuming a single sequences of observed states as the training data, context pattern matching take advantage of the extra (potentially location/area specific) information provided by the segmentation of a GPS stream into a set of trails while still remaining generic prediction algorithms. The logical unit of a trail, while being very generic with respect to the problem of movement prediction, enables a number of reasonable domain-dependent assumptions to be made. One useful assumption that can be made is that trails are independent of each other. This enables the context (and hence context length) to be defined as the preceding states within a trail for any state within a trail. Therefore the context and its length does not need to be estimated as required by the universal prediction problem. Following this approach the known contexts and their corresponding predictions can be used to build a prefix tree and the standard PPM algorithm used, utilizing the fallback mechanism for matching previously unseen contexts. Alternatively partial pattern matching can be applied for which a large body of research exists with respect to dealing with noisy data. In contrast to Universal Predictors partial pattern matching is able to utilize extra information encapsulated in the logical unit of a trial, namely how many states to predict into the future when predicting a route rather than the next step. This is achieved by using the remainder of a matched trail as the prediction. In contrast Universal Predictors predict routes by simply perform next step

predictions recursively until a user-defined stopping condition.

Specifically partial pattern matching with respect to movement prediction can be considered to be a three stage process:

1. *Computation of typical patterns:* A set of typical trails $\mathcal{K}'$ is created from the training data of historical trails ($\mathcal{K}$). The result is a set of unique trails which are considered to be significant in the data set, in other words, they have not occurred by chance. Optionally each pattern additionally contains meta-data indicating how typical the pattern was in the data set.

2. *Computation of best partial match:* Given the pattern set $\mathcal{K}'$, a (partial) input trail E and a similarity function $\delta(\cdot,\cdot)$ determine the closest match as $m = max(\delta(E,K)), \forall K \in \mathcal{K}'$

3. *Prediction:* Since all patterns in $\mathcal{K}'$ were typical patterns the continuation of the matched pattern is used as the prediction. If multiple patterns were matched, and the patterns contained meta-data with respect to how typical that pattern was, the most typical pattern is selected.

Framing the PPM algorithm in this light we see stage (1) is achieved when the algorithm identifies contexts. The historic patterns in this case are the collection of contexts and the state which occurred next in the sequence. The probability of the context and the state is the meta-data indicating how typical the pattern was. Stage (2) is then performed utilizing the prefix tree, an exact matching function and fallback, matching against the contexts. In stage (3) the state which occurred after the next sequence with the highest probability is selected.

The advantage of the context pattern matching is the ability to deal with noise in both identifying typical trails and in the matching phase. For the former a body of work based on frequent pattern mining within databases has emerged with respect to trajectories (e.g. [69, 76, 139]), extending traditionally data mining techniques. With respect to the latter a number of approaches have been proposed and are discussed individually below. In addition, as previously mentioned, a route rather than a next step prediction is readily available.

Matching via prefix trees using custom distance functions: The mining of frequent trails from a data set of movement logs was proposed in [139]. The process mines trails as sequences of edge transitions between locations. Each frequent pattern

was given frequency metadata in the form of a count of the number of times the pattern appeared in the database. For a prediction a prefix tree was constructed from the frequent patterns. Matching was then performed against the tree starting at the root and matching nodes down the tree by one of three matching (distance) functions. The first required exact matching, returning no prediction if an exact match could not be found. The second simply reflects a first order Markov model by reducing the input to the last seen state. The final matching function weighted the confidence of the match by the relative coverage of the input on the frequent pattern. After matching the matched section of the pattern was considered as the antecedent of a rule within the database and the remainder the consequence. Utilizing the frequency counts associated with the nodes of the tree the confidence of the rule was calculated as the similarity function. Comparing the three methods the authors show the benefit of using the full history over a simple first order Markov model, but indicate poor results when using the re-weighting strategy. Such a result lends itself to the conclusion that noise reduction strategies can negatively influence predictions. This is of course possible, however, it is of note that relative coverage is a somewhat ad-hoc strategy heavily reliant on the initial quantization. Of more interest are distance functions that operate in the two-dimensional spatial domain, as later proposed in [137] although limited comparison was performed with other techniques in either case.

The proposal of [137] also mined frequent spatial patterns but additionally included transition times between locations. In order to perform matching, a prefix tree was constructed using the frequent patterns based on the location data. Encoding transition times, edges in the tree were given minimum and maximum values. These values encoded upper and lower bounds on the transition time between the states. Matching was again performed via a custom matching function. The proposed function first attempts a direct match by starting at the root of the tree and selecting nodes exactly equalling the input and fitting in the temporal range specified by the edges. If such an exact match is not possible the distance function first relaxes the temporal conditions penalizing the similarity score. If a match is still not found the spatial constraint is relaxed and the weighted time and distance differences used along with the support of the matched trail to calculated a similarity measure.

Both proposals in [137] and [139] can be considered modified versions of PPM, with both approaches utilizing prefix tree data structures and using conditional probabili-

ties[9] as the base measure when ranking prediction choices. Differing from PPM both define contexts via frequent pattern mining aiming to reduce the effect of noisy data and the problem of overfitting by presenting only relevant patterns to the pattern matching algorithm. Additionally both proposals include custom, domain based, similarity functions which enables fuzzy matching in a spatio-temporal fashion not possible under the universal prediction definition. This again helps address the problems associated with noisy input data. In contrast to [139] and other previously presented approaches the introduction of the temporal meta-data in [137] shows the encoding of previously unconsidered information. However, the extra information is added in a coarse-grained fashion (a range) and as a secondary matching condition with the in-sequence location data requiring to be matched first before the temporal distance is considered. As such the approach is prone to noise in the form of gaps in both the historic data and the input sequence, although this was of negligible concern as originally proposed due to the coarse level of spatial quantization employed. In fine grain pedestrian modelling from GPS logs such occurrences are highly probable with a symbol missing 7.1% of the time on average in the dataset examined in this thesis when quantized spatially by a 5×5 metre grid.

Matching via alternate data structures: Moving away from the PPM paradigm and prefix trees a pattern matching approach called *alignment prediction* is proposed in [178]. Successfully applied in computation biology to find approximately matching patterns between RNA or DNA sequences, the approach aims to address problems of noisy data. The approach addresses the issue of noise in the form of gaps, computing a partial match score via an alignment approach for each typical trail. Using an arbitrary distance function, and allowing parts of both the input and the typical trail to be skipped based on a penalty function, the input is aligned optimally to the typical sequence. The unmatched remainder of the trail with the best partial match score is then used as the prediction. Computationally complex in naive implementations, the approach is made efficient using integer programming and fast approximation algorithms.

Alignment prediction offers a flexible framework for modelling movement prediction and is the closest related work to the approach presented in chapter 6. However, while

[9]Confidence is the name given to conditional probabilities within the data mining community. Due to the tree structure, support (pattern counts) and confidence are proportional. Therefore, since the absolute values are never used, the utilization of support over confidence is an arbitrary decision with support preferred for computational simplicity.

enabling a relaxed notion of sequence, control over the relaxation is via a somewhat non-intuitive skip function. The first author discusses the choice of this function in [177], specifying a function by the assumption that a gap symbol is considered a missing symbol rather than one recorded in error. While such an assumption is highly likely, the alternative is also possible. In chapter 6 this is further discussed and a new approach proposed addressing this issue while maintaining lower worst case complexity bounds.

Principal & Independent Component Analysis: Algorithms in previous sections have all addressed the matching using specific data structures to enable efficient computation, although the basic form of pattern matching simply requires all typical patterns to be compared with the input using a similarity metric [69]. This, however, is computationally prohibitive for large datasets requiring at least $O(|E|n)$ comparisons where $|E|$ is the length of the input and n is the number of observations. An alternative approach that has been proposed recently [56] in the context of high level behaviours is principal component analysis (PCA) and the conceptually similar independent component analysis (ICA). Principal component analysis based approaches have sought to reduce the dimensionality of a dataset while still maintaining as much information as possible, in other words to identify the most discriminative features. The underlying notion is that the distance between a smaller set of the most discriminating features will provide more accurate matches, avoiding the curse of dimensionality. The curse of dimensionality refers to the problem of distance functions becoming less discriminative as the number of dimensions increase [19]. Techniques involving PCA or ICA start by encoding the states as binary feature vectors with each feature becoming a dimension thus resulting in a set of high-dimensional data. E.g. in a world with three discrete locations the feature vector a person located in the second location would be encoded as [010]. At a relative abstract level, and in the presence of multiple context features, good results have been reported [56]. However, at the lower level of pedestrian prediction where the binary encoding of the occupancy of the locations (e.g. $5m$ grid cells) would

result in a huge number of binary features the performance degrades rapidly[10]. This is primarily due to the PCA (and ICA) treating the dimensions as independent when they are clearly highly dependent. Since without throwing away the sequence information (the dependence information lost when using PCA or ICA) the problem is of low dimension anyway and such techniques are not required. As such the problem of pedestrian prediction is not amenable to such analysis, although GPS data could easily form one dimension of a larger set of contexts to which PCA or ICA was applied.

Summary

Context pattern matching addresses the obvious challenges of pedestrian route prediction from GPS logs. The approach provides an intuitive framework to incorporate the domain knowledge encapsulated inherently by the way trails are segmented. Additionally, noise is dealt with in a general way through the use of similarity functions allowing all the historic context present in each trail to be used while still remaining tractable. The benefits, however, come at a price of increase complexity. This complexity is both in terms of computation which, while tractable, may exclude their use in certain situations and in the requirement to choose similarity functions and trajectory alignment heuristics. This has lead to a vast variety of approaches proposed. The most common approach has seen the use of a prefix tree for matching, similar to the PPM universal prediction algorithm, although alternatives in the form of basic nearest neighbour, alignment prediction and principal component analysis have been proposed.

Unfortunately limited comparisons have been made between proposed pattern matching approaches and in particular, no comparisons have been made specifically in the domain of fine grained GPS trails utilized for pedestrian route prediction. A significant number compare only variants of their own technique [69, 137, 139, 200**?**], utilize either unique datasets or the Wi-Fi dataset also used in [182]. As previously discussed the Wi-Fi data is coarse-grained and has vastly different qualities, leading to results in [182] and [31]

[10]Non-binary encodings are also problematic as they require a fixed number of samples as PCA and ICA require fixed length feature vectors. While such a requirement can be met by resampling techniques (e.g. those considered in [142]) the whole process still maintains little merit since the motivation to reduce the dimensionality is limited. Additionally the interpretation of the eigen vectors of such a non-binary encoding has limited merit since principal components become averages of location data which results in the data being represented as a combination of spatially averaged trails. After reducing the dimensionality the resulting trails then provide minimal insight for prediction purposes.

indicating the superiority of low level Markov models with fallback (PPM) compared to higher order models. As perviously noted, this is not a result that is expected to generalize to the conditions of pedestrian movement prediction.

Raw GPS data is utilized in [178] but sampled at twenty minute intervals, reporting superiority of first order Markov models over the alignment prediction in the short term and vice versa in the long term. Note that the use of twenty minute intervals is too sparse to enable pedestrian route prediction. In contrast GPS data quantized into regions of interest by clustering is used in [202] along with a pattern matching approach based on a custom heuristic. The authors compare their approach to a basic Markov model showing a vast improvement. However, the decision to use the average prediction length from their own algorithm as the number of future steps to predict using the Markov model potentially has a large negative effect on the performance of the Markov model. Any missing or extra predicted points occurring due to this choice of length would result in a penalty under the evaluation metric used, artificially degrading the Markov model performance. This issue and alternative approaches to evaluation in the case of comparing fix length and next step prediction algorithms is further discussed in chapter 3. Overall, limited comparative studies exist and none cover fined-grain movement prediction for which it is unlikely that low order Markov models will perform well in general.

2.2 GPS Data: Encoding for route prediction

The route prediction algorithms typically do not work directly from the raw GPS logs. Indeed, the route prediction problem as defined in this thesis in section 1.1 already assumed the segmentation of the GPS data into trails as a pre-processing step in order to separate the prediction problem from the definition of a route. Additionally it was noted that this pre-processing step enables the addition of detailed domain knowledge while directly applying generic prediction techniques (see section sec:patternMatching). Segmenting the GPS logs into trails is not the only form of pre-processing typically performed with many of the previously discussed prediction techniques encoding the GPS point observations in order to apply state-based techniques, reduce the computational complexity of matching and/or in an attempt to deal with noise present within the raw GPS data.

This section consists of three main subsections. The first is focused on the requirement, motivations and methods used to encode the raw GPS data into various forms with respect to the problem of pedestrian route prediction. The second provides an empirical look at the noise present in the movement data set used in this thesis and considers the effect the noise present has on the encodings. This is important as it has the potential to effect a number of the previously discussed prediction algorithms, particularly those that assume accurate sequence information such as Markov models. Importantly this informs the development of new algorithms later in this thesis. The third section address the different issue of segmenting trails.

2.2.1 Encoding location from raw GPS data

The motivations to encode raw GPS point observations are varied. They include the desire to address issues of noise to better encapsulate a notion of matching or similarity, the desire to utilize pre-existing algorithms which operate on discrete or symbolic data, the desire to reduce computational complexity and the desire to reason with locations understandable to the end user. The discussion on the possible encodings is split into two subsections. The first uses quantization in order to address the desires for discrete locations and the second considers the direct use of continuous coordinates.

Encoding via quantization

Encoding to a discrete set of locations via quantization is intuitively appealing as it has the potential to address issues of noise, enable exact matching by symbol comparison (providing computational benefits), enable the direct application of certain algorithms and provides the opportunity to use human identifiable locations. Quantization can be performed individually for each dimension of the data (e.g. location, time) resulting in a state being represented by a vector or globally resulting in a single symbol representing each state. Note that each unique combination of values the vector can take (since each element in the vector is discrete this is a finite number) can be thought of, and mapped to, a single symbol. Hence for the remainder of the discussion quantization of a location is considered to result in a symbol and the quantization of trails considered to be a set of symbols. In any case, in practice, exact comparison between vectors/symbols is cheap since the vector only contain discrete symbols which can simply checked for equality.

In all these cases the process of quantization must take into account the questions *what constitutes a location?* and *when are two locations equal?* or *how close is this location to another?*. Additionally the issues of noise must be considered, particularly if exact matching is going to be used. This is because even if it could be decided at what distance or otherwise two GPS points became *different* there is the additional issue in the technology correctly identifying the device's actual location. The level of noise caused by this GPS inaccuracy is affected by a large range of factors including satellite positions, signal obstructions (such as buildings), quality of the device, atmospheric conditions and signal interference known as multipathing where signals from the satellites may reflect off an object and take an indirect route to the receiver [89]. Note that if this is not taken into account and very fine-grained quantization used exact matching will generally fail and no predictions will be made. On the other hand if a sufficiently coarse-grained quantization is used the effect of small changes in the raw GPS readings due to error can be removed.

A final point to note when encoding GPS data for use in algorithms using exact matching is how time is quantized. In general time is converted into sequence after location is quantized to a symbol representation (e.g. for use in Markov Models and Universal Predictors). This form of quantization leads to issues when comparing trails since missing points in the raw GPS data mean that the trails do not align as would be expected. Missing points can occur for a variety of reasons including differing sample strategies (i.e. different reporting requirement across different devices requiring them to report at different temporal or spatial intervals.), filtering algorithms (i.e. those that remove impossible samples based on velocity and distance thresholds and the laws of physics), device error and/or temporary occlusion of satellites.

In the remainder of this section previous approaches to encoding GPS via quantization are discussed considering their applicability to the problem of pedestrian route prediction.

Logical location matching: Logical location matching has been employed in two main cases (1) in the transport domain segmenting roads at intersections to construct a complete set of possible locations, and (2) when predicting movement between known regions of interest (RoIs) such as train stations and specific tourist destinations. In both cases exactly the same techniques are applicable as the task is simply one of mapping points of continuous latitude and longitude (and potentially other features) to the sym-

bols in the domain of known locations. The techniques range from simply using fixed regions based on domain knowledge (e.g. Wi-Fi zones [31, 147]) or employing heuristics such as correcting the point to the road with the smallest distance and angle deviation from the previous adjusted location [200] to utilizing a hidden Markov model and the Kalman filter [114, 116] or hierarchical Bayesian networks [149]. With respect to pedestrian route prediction, knowledge of a fixed set of possible locations can not generally be considered to be available. Pedestrians are not constrained to roads and the quantization to regions of interest removes the route information. As such these approaches are not considered further.

Clustering: Clustering in this context can be considered a task of automatically identifying either a set of regions of interest or a complete set of possible logical locations. One of the first proposals was the use of a modified, iterative k-means clustering algorithm [11] which required a radius specification rather than the specification of the number of clusters. In a similar fashion [95] propose the use of the DBSCAN clustering algorithm separately at each time point. This has the benefit of not assuming that locations are represented by circular areas but requires observations to be time-aligned. Density based approaches were also proposed in [76] and variants utilized in [137, 202]. The first proposed approach in [76] quantized the spatial area via a grid. Square regions that were visited (optionally including through interpolation) by at least a fixed number of trails in the data set were then selected using a density threshold in a similar fashion to DBSCAN. Having extracted a set of *popular* fine grain locations a heuristic was used to identify regions of interest. The second approach proposed selecting all crossroads where more than a certain number of the crossing trajectories change their direction, although this technique was not elaborated on or evaluated. While computing RoIs are not of much value in route prediction, the selection of a reduced set of dense grid squares can help computational tractability, although otherwise simply reflects a uniform grid quantization approach. The investigation of cross roads may be of worth, but may also result in throwing away useful information and has not been investigated in this thesis.

Uniform quantization: The simplest approach to reducing noise and creating discrete places for reasoning is via grid based quantization. In this approach the aim of identifying locations with specific semantic meanings is somewhat abandoned and the motivation mainly due to increased practicalities when performing matching (reduced computational effort) and modelling (amenable to discrete prediction models). A further motivation is

that the approximate spatial error rate of the GPS devices is generally known providing a reasonable estimation of the grid size required to get continuous sequences in general reducing the issues of sequence alignment due to missing GPS point observations. Finally it is of note that after grid based quantization the grid locations can either then be used in distance functions (via their centroid or grid reference) to help address the problem of incorrectly sequenced symbols, or as a set of discrete symbols.

Direct use of continuous coordinates

In the approaches discussed noise has been dealt with as a pre-processing step, taking noisy continuous data as input and providing a set of sequences of discrete symbols generally treated as noise free as output. In contrast [69, 178] employ a similarity/distance function when comparing trajectories containing continuous valued sample points. Due to the noisy nature of the raw data both similarity functions provide a mechanism to deal with noise with [69] matching points to line segments with the drawback of high computational complexity. In contrast [178] propose the use of alignment prediction techniques which has a similar worse case complexity but for which efficient implementations and fast approximations exist. The direct use of a distance function has the added advantage that it provides a higher level of discrimination than exact matching and can determine the *closest* trail in cases where exact matching cannot. The method proposed in chapter 6 extends the direct use of continuous coordinates to exploit this extra discrimination additionally enabling a direct approach to model sequence and time while providing improved worst case complexity.

2.2.2 Properties of noise within a data set from mobile phone GPS data

The previously discussed encodings and the corresponding prediction algorithms deal with noise to varying degrees. In this section movement data sets used within the literature are first discussed where it is noted that limited information with respect to noise levels is available. Due to this the data set used in this thesis is examined to provide some insights into the levels of noise in this type of data in order to inform the development of the prediction algorithms presented later in this thesis.

Data sets used for movement prediction have generally recorded data at either the in-

dividual level or at a transportation level, e.g car or taxi. At the transportation level GPS is almost exclusively used [69, 116, 137, 200**?**]. At the individual level data sets can be further categorised by the level of accuracy of the technology recording the location. The three levels are mobile cell tower traces [22, 55, 56], Wi-Fi access point logging [147, 178, 182] and mobile GPS data [11, 118, 178, 202, 203]. A final category is synthetic data (i.e. as used in [95, 139]) which is not focused on in this thesis. Clearly each type of data has a variety of properties. For example data recorded at the cell tower level is unlikely to have missing data leading to sequencing problems, transportation data is likely to be more accurate depending if the GPS device performs some map matching automatically and Wi-Fi access can only be invoked where access points exist and a user chooses to connect. For pedestrian route prediction the datasets of interest are GPS based data sets at the individual level. Most data sets have been characterized by a small number of users (e.g. studies with 1 - 6 participants [11, 118, 178]) over varying time periods from six days to seven months. Recently, however, work on larger datasets have emerged with [202] involving 17 participants in one month resulting in 900 trips. Additionally work using a dataset from Microsoft research Asia has recently been published [203] although the work focused on location and activity recommendation and as such was focused on a much higher level use of the GPS data. In all cases only limited descriptions of the data is present and limited discussion about the levels of noise.

The dataset used throughout this thesis is a called the *D-SCENT* dataset. The D-SCENT dataset was generated from an augmented reality simulation that was developed as part of the D-SCENT project funded by EPSRC at the University of Nottingham in the UK. Participants took on the role of workers constructing an Olympic site, performing a host of purchasing and building tasks. It is important to note that while tasks undertaken by participants were artificial, their movement across the real world playing area was completely unconstrained and reflects real spatial behaviour. The dataset was collated over a year, with the simulation featuring 12 locations and covering a $80,000m^2$ spatial area. Sixty participants interacted with the game via G1 smart phones and their (assisted) GPS data was collated every 5 seconds by a central server resulting in over 30,000 position records. Therefore the data set represents a dense collection of relatively short term movement patterns, by a multitude of individuals, between a number of controlled locations, under real world noise and movement with goals unknown to the system as a result of the game rules and team dynamics.

In section 2.2 two main categories of noise were noted in the case of discrete data. These

were errors in the position reported and missing GPS point observations. The first leads to symbol corruption where incorrect symbols are recorded given the true position of a person. The second leads to symbols not being included (missing symbols) resulting in sequence alignment issues when comparing trails. Since symbol corruption is impossible to measure in general and is indistinguishable from subtle movement deviations which may contain discriminative information for the predictions, therefore this noise type is not examined other than to note that the percentage of pairwise matches between trails (segmented at known destinations) that exactly match is very small, only 0.0467% at ten metre quantization and 0.0006% at five metre quantization. If the notion of equality is relaxed to one trail merely containing all locations of the other in any order then for ten metre quantization this improves drastically to 0.9963%, to over the number expected if there was only one possible route between each of the twelve locations ($\frac{1}{12(12-1)} \approx 0.7576\%$). Under five metre quantization, however, the number is still comparatively low at 0.0789%. In all cases trails were quantized using known locations.

The second category of errors is missing symbols (after location quantization). Missing symbols relates to uneven sampling. This often occurs in real world data logging presenting a problem to algorithms assuming correct sequence information. Assuming that under quantization a continuous path is expected, this can easily be examined providing an insight into the number of missing symbols. Again trails were created using a known set of destinations. The number of missing symbols in each trail was determined by counting the number of symbols that were not directly followed by another in the symbols direct spatial neighbourhood. This was then averaged over the length of the trail providing a measure representing the probability of a missing symbols, per symbol in that path. On average the probability of a missing symbol, per symbol was 1.9% with a standard deviation of 5% under ten metre quantization and 7.1% with a standard deviation of 7.9% under five metre quantization. In other words, under ten metre quantization approximately 2 in 100 symbols were missing. Under five metre quantization this jumped to around 7 in 100 symbols being missing. In the D-SCENT data, ten metre quantization produced trails with a median (mode) length of 8 (6) spatial regions with an interquartile range of 5. Five metre quantization produced trails with a median (mode) length of 13 (10) spatial regions with an interquartile range of 10. Histograms of the distributions are shown in figure 2.7.

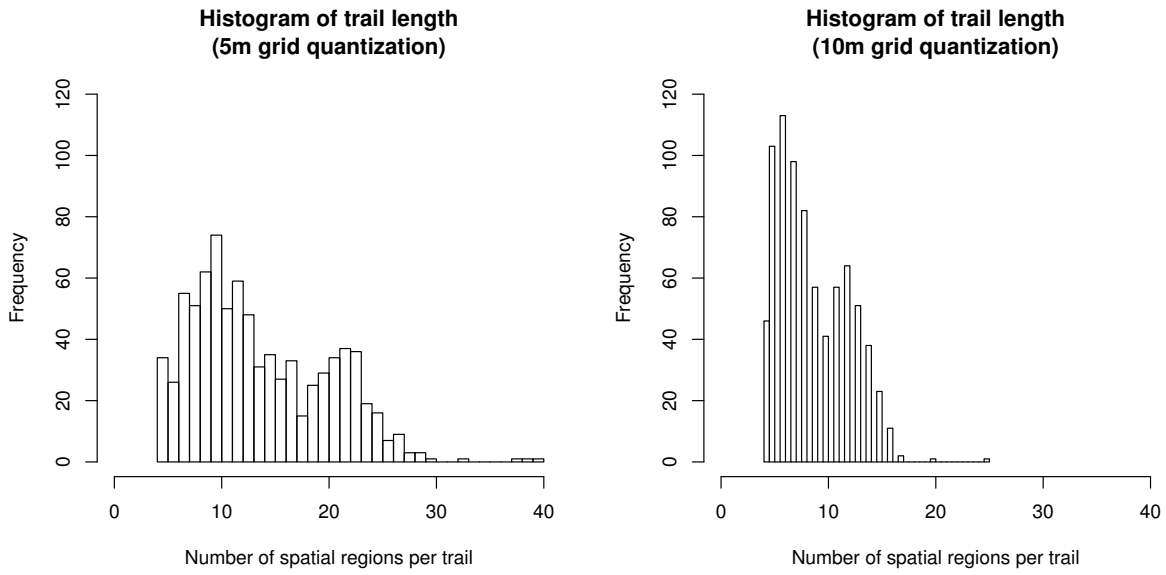


Figure 2.7: Histograms showing the trail length distribution for trails from the D-SCENT data set quantized using a uniform grid of 5 and 10m. Trails were segmented based on a set of known locations.

2.2.3 Trail identification

As previously motivated the route prediction problem considered in this thesis requires the identification of trails from the raw GPS data. These trails simply represent logical movement patterns (routes) and provide a domain specified limit to the context and prediction horizons within prediction algorithms. This aids in reducing computational complexity and can help provide more accurate predictions. Most importantly it helps separate the definition and subsequent identification of routes from the prediction mechanisms. This allows this thesis to focus on the latter.

The most common approach to trail identification is to utilize a heuristic that segments trails based on a threshold of inactivity or signal loss [132]. This represents the notion that a location is one in which a user either enters a building (signal loss), pauses or puts down the device at destinations (such is the case if the GPS receiver is in a car [69]) and that these locations should be used for segmentation. The heuristic is therefore based on both a temporal threshold between observed datapoints (e.g a three minute threshold is used in [69], ten minutes in [11]). Trips are then typically filtered heuristically to remove those where minimal movement occurred under the assumption that such movement was captured in error. Alternative approaches seek to learn the destinations [119**?**]. The simplest, however, is a pre-known set of destinations, although this is generally not

available.

2.2.4 Summary

In this section both the challenges associated with using incidentally collected GPS location data and the challenge of then encoding this data such that matching/comparisons could be performed was examined. Prior work has utilized either discrete encoding and exact matching based on these discrete states or distance functions either on the continuous coordinates directly or after discrete encoding. In the work reviewed a number of different approaches to quantization were discussed. Of these clustering is not considered at all in the thesis and logical locations are only used to consider the noise within the data set. In the case of logical location matching this is due to the unrealistic manual effort required in general. In the case of clustering this is due to the approaches in the literature addressing the problem of identifying regions of interest which then prevents the generation of routes.

The remaining approaches to encoding GPS location data reviewed are uniform quantization and the direct use of continuous coordinates. It is of note that the utility of these encodings can be vastly different, with uniform quantization allowing prediction algorithms based on discrete encodings. As previously discussed using discrete encodings can significantly reduce the computational complexity of the predictors by only performing exact symbol matching. Such a decrease in computational complexity generally also comes at a cost with respect to the accuracy of the prediction algorithm, but in some cases such a trade-off is acceptable. Such an algorithm is motivated and developed in chapter 4 where two different levels of uniform quantization are used and compared. In contrast, in chapter 6 the continuous coordinates are directly used in a novel prediction algorithm developed to achieve the highest prediction accuracy.

An interesting aspect not covered in this thesis is the empirical comparison of the effect of quantization and the direct use of coordinates under identical algorithms. Such a comparison is left as future work.

Following the review of approaches to encoding the GPS location data the properties of noise within the GPS data set used in this thesis was examined providing insights into the amount of noise present, highlighting such a data source is noisy even when trail quantization is performed with a set of known locations. For instance in the data

set used in this thesis 98% of the time at least one symbol will be missing per trail if quantized with a five meter grid.

Finally methods for trail identification were discussed briefly for completeness, as while such methods are used in this thesis, the extension of such methods is beyond the scope of this thesis.

2.3 Conclusion

In the first part of this chapter a large number of prediction algorithms applied in a variety of movement prediction contexts were examined considering the general challenge in developing prediction algorithms for pedestrian route prediction. Unfortunately, while many approaches to movement prediction exist, no comparative study using fine-grained GPS data exist for pedestrian route prediction, with the closest study being that by [178], however, the sampling rate used was very coarse with samples taken at 20 minute intervals. Other studies either use data sets other than GPS data that is of a much higher granularity (e.g. the comparative study on Wi-Fi data in [182]) or first detect stay points or regions of interest (e.g. [202]) or simply only compare variants of their own algorithms.

In this light the chapter provided a theoretical summary and evaluation of the potential of algorithms in the field of movement prediction, providing a broad but detailed summary of related work in movement prediction approaches in general. Specifically the chapter provided a critique of the approaches in line with the challenges of utilizing the full historic context and the issue of noise which are particularly relevant to pedestrian route prediction from GPS logs. The general conclusions are summarized below, following which a number of algorithms are highlighted as representative baselines for the empirical evaluations of the proposed algorithms in chapters 4 and 6.

While a large number of specific algorithms were discussed here they are grouped into seven categories: fixed order Markov models, Markov decision processes, hidden Markov models, Kalman Filter, hierarchical dynamic Bayesian networks, universal predictors, pattern matching approaches.

Of these fixed order Markov models, Markov decision processes, hidden Markov models and the Kalman Filter are unlikely to adequately utilize the long context information

which is present in real world GPS logs. While hierarchical dynamic Bayesian networks have the potential to utilize longer historic context, allowing powerful representations, they are dependent on a specific hand-crafted model of sufficient complexity to encapsulate the observed multi-time step dependencies. Additionally such models are significantly more complex in both learning and inference, with approximation algorithms required in the case of more complex models. The remaining approaches are all able to explicitly model long context histories.

Of the remaining approaches, PCA and ICA fail to deliver good prediction performance due to the use of an encoding which discards sequencing information. The rest can be seen to have much greater potential.

Universal predictors have good theoretical potential, however, they do not account for noise inherently, relying on an external method of feature encoding. In addition they do not provide mechanisms to handle varying sample rates. Note, however, that by using data encoded to discrete state symbols and exact matching the computational complexity of the algorithm is significantly reduced.

Adopting a different paradigm, pattern matching approaches provide a way of both directly modelling higher order sequential data thereby utilizing all historic context while also addressing issues of noise though similarity functions. It is in this highly flexible framework that many of the most recent approaches to movement prediction have been realized, due to the large number of different heuristic matching and similarity functions that can be applied. A drawback to these approaches is the computational complexity which can vary significantly once again due to the specific matching and similarity functions used.

Noting this large difference in computational complexity motivates the development of the two prediction algorithms presented in chapters 4 and 6. The former focuses on maximizing accuracy while still being able to run on very limited hardware while the latter is focused solely on performance and general tractability, additionally motivated by analysis in chapter 5. The lack of empirical comparisons with respect to pedestrian route prediction motivates chapter 3 which considers the evaluation of predictors and requires the selection of a number of baselines predictors. In all cases the prediction algorithm from [137] is selected. The algorithm can be classified as a pattern matching approach and represents a recent, state-of-the-art approach in the general field of movement prediction. When used as a baseline the approach is modified slightly to enable fine-grained

route prediction, by simply skipping the step where the GPS streams are quantized into regions of interest. In the evaluation of the performance focused algorithm proposed in chapter 4 two predictors of similar computational usage are included as baselines. These are a standard first order Markov model and a variable length Markov model representing the class of simple models and universal predictors respectively.

The second part of this chapter focused on challenges associated with using incidentally collected data, namely the issues of noise and missing symbols. Here an empirical investigation of the data set used in this thesis provided evidence confirming the theoretical expectation that the GPS data used in pedestrian route prediction is both noisy and contains missing symbols. Good prediction algorithms in the context of prediction from GPS data should therefore take into account that missing symbols regularly occur in addition to considering errors in the reported location. Additionally discussed was challenge of encoding the data. Specifically the decision to convert the data into discrete symbols via quantization or not was discussed. Here it was noted that it is dependent on the aim of the predictor, which symbolic comparison significantly cheaper than the comparison of continuous variables via a distance function. Of course the performance in either case is heavily dependent on the prediction technique used and the empirical evaluation of the performance of algorithms under both conditions was noted as interesting future work. Note that quantization via clustering was not considered applicable to the problem of route prediction and is not considered further in this thesis. The remaining previously proposed approaches of grid based quantization and no quantization are further considered later in this thesis in predictors evaluated in chapters 4 and 6. For the former pertains to an approach where the definition of a good prediction algorithm is more related to runtime performance while the latter approach is focused primarily on prediction accuracy. As expected the predictors using no quantization and a distance function to account for noise show an increased performance. Finally, while only utilized in this thesis and not examined, methods for the identification of logical trails within recorded GPS data streams were discussed for completeness.

Chapter 3

Evaluating route prediction algorithms

It is easy to lie with statistics. It is hard to tell the truth without statistics.
- Andrejs Dunkels

Throughout this thesis numerous different algorithms and their various forms of parameterization for predicting pedestrian routes are presented and compared. Clearly such an undertaking requires a robust framework for evaluation. While in many fields there is a standard performance metric, such as receiver operator curves (ROC) for evaluating binary classifiers [64], or precision recall curves in the information retrieval field [163], a survey of recent literature with respect to movement prediction reveals no such standard practices. For instance out of the 16 papers directly relevant to low level movement prediction from logs reviewed in this thesis roughly only a third share a common evaluation metric with another paper. Furthermore only five compare their results to other techniques and of none report any levels of significance associated with their findings, rather only presenting descriptive statistics.

In light of this and to provide a robust foundation for the discussion and comparison of the algorithms presented in this thesis, this chapter provides a unified view of utilized evaluation methodologies within the literature. Based on that, a recommended set of evaluation metrics, visualizations, and statistical comparison procedures is outlined. This forms the basis of comparison used throughout this thesis and provides a set of simple but robust guidelines for others wishing to compare similar algorithms within the

field.

Specifically this chapter examines the evaluation of algorithms addressing the route prediction problem as detailed in chapter 1, section 1.1. For clarity the definition of the route prediction problem is repeated below:

- Given a set of historic trails: $D = \{H_1, \ldots, H_{|D|}\}$ where a trail $H = \langle h_1, \ldots, h_{|H|} \rangle$ and each point is equal to some feature vector (minimally encoding location), for instance: $h = [longitude, latitude, time]$
- And given an observed input trail $E = \langle e_1, \ldots, e_{|E|} \rangle$, where e is a point observation of the same type as h
- Provide a resulting prediction $R = \langle r_{|E|+1}, \ldots, r_{|R|} \rangle, |R| \geq 1$ of arbitrary length indicating the route that is expected to be taken, where r_k is a feature vector encoding only locational information (e.g. $r_k = [longitude, latitude]$).

Recalling once more that H, E and R are all ordered sets as indicated by the angle brackets.

From the problem definition the high-level evaluation of prediction error for a individual prediction can be easily defined as $\delta(R, F)$ where R is the resulting prediction as defined in the problem definition, F is what actually happened and $\delta(\cdot, \cdot)$ is some distance function returning the dissimilarity of R to F. This notion of prediction quality is similar to that defined in [177]. The goal of any prediction algorithm is then to minimize the distance measure across all predictions although it is of note that exactly what is being minimized, for example the average error, is open to debate. The definition of the distance function and the definition of what form of aggregation should be minimized is the first challenge in the evaluation of route prediction algorithms and is addressed in section 3.1.

The second challenge, examined in section 3.2, relates to how the set of historic trails, D, is used to both train predictors and also evaluate them. In practice the whole of set D would be used to train the predictor. However, for evaluation some of the trails from D must be used to construct the input (E) continuation used as the known result, F, against which the prediction is evaluated. Of course when making predictions in the real world F is unknown. However, we can obtain F for testing by taking a full historic trail and cutting it into two segments. The first segment then becomes E (the observed input trail from which the prediction algorithm makes its prediction) and F becomes what

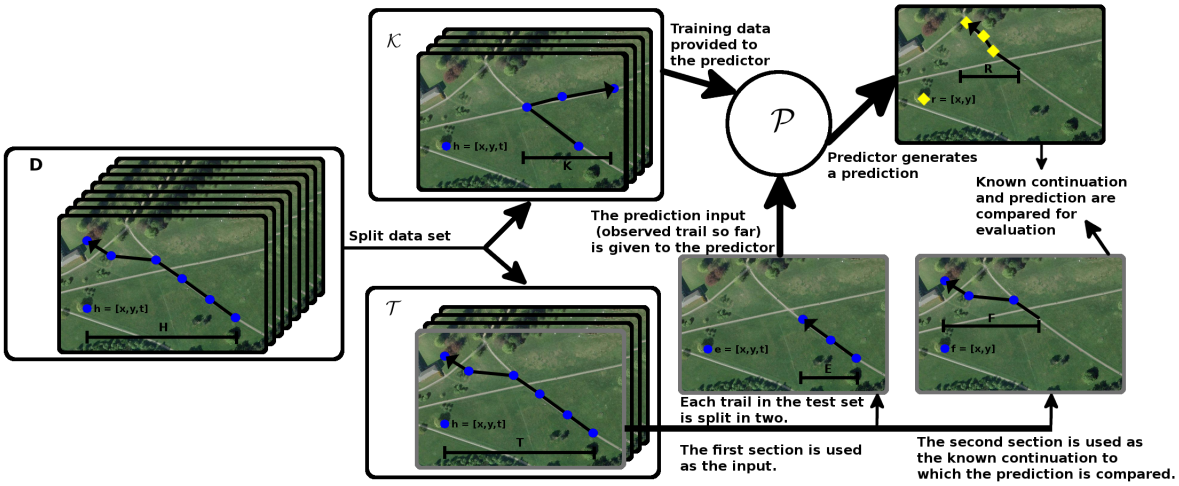


Figure 3.1: Figure visually showing an the process of a basic evaluation procedure labelled with the corresponding symbols used throughout this thesis.

actually happened. Therefore two sets must be made from the data set, D. The first is the training set. This is what the predictor is provided with ahead of time to be used for learning. The training set is denoted throughout this thesis as: $\mathcal{K} = \{K_1, \ldots, K_{|\mathcal{K}|}\}$, where $K \in D$. The second is the test set, a set of historic observations from D which are each split to construct pairs of E and F. These pairs are then used to evaluate the predictors. In this thesis the test set is denoted as: $\mathcal{T} = \{T_1, \ldots, T_{|\mathcal{T}|}\}$, where $T \in D$. The process is illustrated in figure 3.1. The challenge in this instance is the provision of a methodology to divide the a data set of historic trails into these training and test sets. How this is done is important as it can subtly change the type of error being evaluated between three main variants.

The first possible type of prediction error is the prediction error of a *trained* model. In other words, the expected error of a predictor given a fixed training set. Clearly this is not the desirable measure of performance with respect to movement prediction algorithms from GPS data for which the training data has the potential to change rapidly. The second type of prediction error is the prediction error over training sets. This assumes a specific data universe (e.g. a specific problem domain for which data has been gathered) but is a generalization of the prediction performance to an arbitrary training set from that domain. In reality, the calculated generalization error is specific to the size of the training set repeatedly used to calculate the error. The third measure of error is the predictor error over multiple different data sets. This provides a further generalization of the error over a wider range of conditions and inputs and/or domains. Clearly this is the

desired goal in general. Additionally it is somewhat easier to perform statistical analysis with since the different data sets are generally considered independent, a highly desired property which single data sets lack. The implications of this point form the majority of the discussion in section 3.2. Despite the benefits of using multiple data sets it is common that a sufficient number of data sets do not not exist or are not accessible and so the second type of error must be used. This is the error that is focused on in this chapter. From a theoretical perspective this is still a correct performance metric when evaluating an algorithm for real world use in a specific domain. From a pragmatic perspective it is the best possible choice when limited data sets exist, which is the case with GPS logs at this point in time. It is of note that this has generally not been considered in the literature, with the majority of the reviewed papers simply splitting their data resulting in only evaluating the prediction error of the trained model. Of the remaining papers, two used leave-one-out cross validation to obtain the generalization error, one used an unspecified form of cross validation, three did not report their methodology and one resampled from the training data, a technique which is known to produce artificially good results.

A third challenge, directly coupled with the second challenge of how to choose the test and training sets, is the choice of which statistical tests should be used to compare the algorithms and provide evidence as to the validity of the results. Clearly an important issue, this is also examined in section 3.2 arriving at a set of recommended procedures in the case of route prediction evaluation.

Finally it is important to provide descriptive statistics and visualizations. This is addressed in section 3.3 where a concrete set of visual and descriptive statistics are motivated in order to provide a good overview of different aspects of the performance of route prediction algorithms.

In this chapter each section concludes with a recommended approach with respect to the aspect of the evaluation problem they address. These recommendations are then summarized in section 3.4 providing a complete framework for evaluating route prediction algorithms.

3.1 Metrics for evaluating prediction quality

The first hurdle in evaluating predictions is defining the distance function, $\delta(\cdot, \cdot)$, which returns a measure of how different an individual prediction is from what actually occurred. From the problem definition presented in section 1.1 the distance function is of the form $\delta(R, F)$ where $R = \langle r_{|E|+1}, \ldots, r_{|R|} \rangle, |R| \geq 1$, $F = \langle f_{|E|+1}, \ldots, f_{|F|} \rangle, |F| \geq 1$ and $|E|$ is the length of the observed trail from which the prediction was made. This encapsulates the stipulation that route predictions contain order, i.e. that both the prediction and observed routes are a set of data points with a known order. Furthermore it is assumed that this order refers to a timing order with successive points occurring after preceding points. Additionally the subscript $|E| + 1$ denotes that the prediction and corresponding truth follow on from the observed trail. In the remainder of this chapter this notation will be dropped for conciseness, with the start of the prediction/known truth indicated with an index of zero.

Importantly this definition makes no stipulations that the data points are equally spaced in time or correspond to any point in time. Therefore both the prediction and any observed routes can be considered time series of arbitrary lengths over varying time periods with an arbitrary sample rate. It is important to note that in the evaluation it is desired to measure the difference between the route and not the route and timing information. This is implicit in the problem definition which specifically only returns a feature of spatial location omitting time. As motivated in section 1.1 this is done so as to not conflate the accuracy of algorithms with respect to the prediction of a route with the prediction of the timings of the route. In practice what this means is that timing information can not, and should not, be used to interpolate and align the sampled points that are part of the prediction (R) and the known trail (F).

The above discussion highlights the information that is available in order to evaluate a prediction through the definition of a distance function. Of course this does not provide an indication what such a distance measure might be with many approaches possible. Considering this sections 3.1.1 - 3.1.6 provide a survey of different measures used within recent literature for varying prediction tasks. It is of note that in some cases numerous methods have been used within individual papers. Following the survey an alternate distance measure is proposed in section 3.1.7.

3.1.1 Average log loss

A common method for measuring prediction accuracy within the class of universal predictors (see 2.1.3) is the average log loss [134] which is directly related to the likelihood of the predictor predicting the correct sequence. In other words the predictor with the minimal average log loss score predicts the sequence and all sequential sub-sequences with the maximal average probability. It is of note that the log loss does not split a test historic trail into an input and an evaluation section but rather considers the probability of the sequence being modelled by the predictor. Formally the average log loss, with respect to a test historic trail, $H = h_1, \ldots, h_{|H|}$ and $\hat{P}(\cdot|\cdot)$ the conditional probability as learnt by the predictor, is defined as (from [14]):

$$l(\hat{P}, H) = -\frac{1}{|H|} \sum_{i=1}^{|H|} \log \hat{P}(h_i | h_1, \ldots, h_{i-1}) \tag{3.1}$$

The approach evaluates a predictor's ability/likelihood to *exactly* predict the correct sequence. As presented above this incorporates the predictors ability to model how likely the observed trail was as well as how probable the unobserved portion was. This can be seen by recalling that in the context of prediction problem addressed in this thesis it is assumed that H has been split in two representing an observed portion of the trail (E) and and an unobserved portion (F). While potentially undesirable this is less of a concern as one could imagine a reformulation that only averaged the probabilities relating to the symbols contained in F. In either case, however, a strict evaluation of only the exact solution is being made. This is not optimal for identifying predictors with good practical performance in the domain of pedestrian movement where *close* matches can be of practical use. This is in contrast to other domains such as lossless text compression where close alternatives would lead to data corruption and hence are of no use. Additionally the continuous and noisy nature of the movement prediction domain make exact matches unlikely in the general. Therefore the average log loss is not as applicable as other evaluation functions since, within the domain of movement prediction, the best predictors are considered those that make the closest predictions to the correct route, rather than those that make the exact prediction with the highest probability.

3.1.2 Accuracy as a proportion

The majority of studies within the movement prediction domain consider the prediction task of predicting only the next location (data point) [31, 55, 182, 200, 202] or the final destination [147**?**]. In such situations the most prevalent approach has been to evaluate individual predictions in a binary fashion.

This represents the simplest strategy based on the prevalent use of prediction mechanisms which work with symbolic data. As such distance functions are not required and the prediction is considered correct if the prediction and the known resulting symbol are the same and incorrect otherwise. The approach makes the assumption that predictions beyond the exact match are not useful, and that the boundary at which a prediction is no longer *close enough* to be considered correct is equal to the boundaries in place from the symbolic encoding. In the case of region of interest encoding this seems highly likely. In the case of symbol encoding via grid quantization, however, the correspondence is less clear. In either case, explicit consideration is potentially more desirable.

Generalizing this to route accuracies, [116] define a binary similarity function over complete routes with a predicted route *correct* if there was at least 95% overlap between the prediction and the correct route in the sections common to both routes. In contrast [137] define accuracy per individual prediction as the proportion of correctly predicted locations over the individual predicted route. While intuitively more appealing as it does not rely on a somewhat arbitrary level of overlap, the approach requires the alignment of data points in the prediction and the correct route.

3.1.3 Other measures without pointwise distance functions

The Levenshtein distance is proposed to evaluate the quality of a prediction in [202], citing its tradition as a metric for comparing sequences. The Levenshtein distance measures the distance between two sequences as the number of *edit operations* it takes to transform one sequence into another. Allowable operations are insertions, deletions or substitutions of a single symbol. Unless some form of normalization is performed the resulting score is non-intuitive when aggregated due to the varying lengths of trails. Finally as with accuracy by proportion approaches this metric does not take advantage of the fact that the symbols represent (at least) locations which have an associated distance metric and hence are unable to provide an as-specific and discriminative interpretation

of *close* and therefore a more discriminative quality of the prediction.

In [139] a custom *quality* measure is proposed. Considering next step prediction the approach first mines a set of possible predictions to which probabilities are assigned. After ranking the probabilities, the probability exactly matching the correct result is used, diminished by the difference between the probabilities of higher ranking possible predictions. Again the approach does not use a pointwise distance metric, rather relying on symbolic matching and generalizing such an approach to route prediction would have similar issues with respect to movement prediction as average log loss.

3.1.4 Accuracy as average error, RMSE & MAE

With location data there is almost always a distance function available over the individual location measurements. In the case of destination or next step prediction by using a distance function the accuracy can be measured as the distance between the prediction location and the reality.

In the case of route prediction distances between aligned point pairs along the prediction and the correct route can be taken. Two common metrics used to measure the difference in time series are Root of the Mean Square Error (RMSE) and the Mean Absolute Error (MAE). Given a time series of n time steps, the prediction $R = r_1 \dots r_n$ and correct answer $F = f_1 \dots f_n$ the RMSE and MAE are defined respectively as:

$$RMSE = \sqrt{\frac{\sum_{i=1}^{n} (r_i - f_i)^2}{n}} \tag{3.2}$$

$$MAE = \frac{\sum_{i=1}^{n} |r_i - f_i|}{n} \tag{3.3}$$

Both the RMSE and the MAE metrics adjust the score to take into account the varying length of the predictions by dividing by the length of the time series. Additionally both provide a mechanism to remove the sign of the error so that the magnitude of the errors are what is being measured. In the case of RMSE this is achieved though squaring the differences. In the case of MAE this is achieved by taking the modulus. Of the two the MAE has the simplest interpretation, being the average of the error between each set of aligned points in the prediction and answer. In terms of trails with only location information it is the average distance each point in the prediction was from the correct point in the answer for the same position in the sequence. In contrast the RMSE has a more complex interpretation. Of note is that, compared to

MAE, large errors between individual points in the sequence have a relatively greater influence on the total than smaller errors. Considering only the total square error part of the RMSE (the $\sum_{i=1}^{n}(r_i - f_i)^2$) it is clear that this will grow as the total error is concentrated within a decreasing number of increasing large individual errors. The division and square root then only scale the value and hence this is a more complex notion of error which is additionally based on the distribution of the error. As an example consider two sets of prediction/truth pairs $a : R^a = [10\,100\,100], F^a = [0\,0\,0]$ and $b : R^b = [10\,10\,190], F^b = [0\,0\,0]$. In both data sets the average error (MAE) is the same. In contrast, since in data set b the difference is more concentrated in a single observation, a higher relative score is reported by the RMSE for data set b compared to data set a. Specifically $RMSE(a) = \sqrt{\frac{10^2+100^2+100^2}{3}} \approx 81.85$ vs. $RMSE(b) = \sqrt{\frac{10^2+10^2+190^2}{3}} = 110$. A more in-depth comparison of the MAE and the RMSE can be found in [197].

While both the RMSE and MAE provide a distance measure for series of length n it is of note that the prediction may be longer or shorter than the correct prediction. Additionally the metrics require that the time series be aligned. Since only sequence information is known, and in any case it is desired to measure the spatial similarity only and not the spatial and temporal similarity, resampling uniformly to align the sequences based on time is not possible. An alternative would be to resample based on distance. However, the two-dimensional nature of locations mean that this quickly leads to unintuitive results. This is discussed in more detail in section 3.1.6 and highlighted through the MAE example in figure 3.3.

3.1.5 Hausdorff distance based metrics

Rather than re-sampling and using a standard metric such as the MAE or RMSE the authors in [69] and [185] define measures that takes two trails as the base level of input rather than a set of points. The similarity between the two trails is then calculated via Hausdorff Distance based algorithms. Broadly speaking these measures report the average minimal distance between the prediction (R) and the known trail (F) for each point in each trail.

[185] propose that two trails should be considered the same if the maximum distance between each point in one trail and the closest point in the other trail is less than a pre-specified amount. The approach is asymmetric in that, given two trails, e.g. a prediction (R) and corresponding truth (F), the distance between R and F does not

equal the distance between F and R. Accounting for this two pre-defined maximum distance values are required, one for each case. The authors propose that these values should be the largest sampling intervals in the trails. The use of such an approach as a full measure to adequately calculate the relative distance between trails, however, is not discussed. In this case pre-defined thresholds can not be used but rather an average could be taken over the two results. This approach still suffers slightly from the effect of sample rates. It is of note that, while not discussed, the approach is corresponds to calculating the Hausdorff distance (which is not symmetric) for $\langle R, F\rangle$ and $\langle F, R\rangle$ and applying a threshold to each result. The Hausdorff distance between to (finite) sets of points is defined as:

Recall: $R = \langle r_1, \ldots, r_{|R|}\rangle, |R| \geq 1$ and $F = \langle f_1, \ldots, f_{|F|}\rangle, t \geq 1$,

Then:

$$hausdorff(R, F) = \max_{r \in R} \min_{f \in F} \|r - f\| \tag{3.4}$$

An important weakness of the Hausdorff distance in this setting is that it does not take the order of the points within the curves into account. Intuitively the problem is that points in one trail can match in an arbitrary order against the other. This can lead to small Hausdorff distances when much larger ones would be expected when comparing trails. This is highlighted in an example in figure 3.2 (a).

Aiming to directly address both the sample rate and issue of ordered points [69] define a new distance measure. To prevent confusion this distance measure is referred to as the Froehlich-Krumm distance in the remainder of this chapter. The Froehlich-Krumm distance computes the average of the shortest distance between points in one trail and line segments in the other. Additionally the condition that the segment matched to must be either the same as the previous match or occur after the one selected by the previous match. This acknowledges that the trails contain ordered points. However, as shown in the example in figure 3.2 (b) it does so in a way that may not be intuitive. Specifically the Froehlich-Krumm distance calculates the average of the distance between each point in the first trail and the closest line segment in the second trail and then vice-versa averaging the results. In each asymmetric distance calculation not all points are required and as such measurements between points are not made as would be expected from a global perspective comparative to either a good alignment via good sampling (as shown

in figure 3.2, as used by the MAE) or via global compuationally determined alignment (as done by the Fréchet distance discussed later in section 3.1.7). The requirement that not all points are required in each distance calculation coupled with the nature of the minimum function tends to lead to underestimating the distance compared to what would be expected. A formal definition of the Froehlich-Krumm is provided in equation 3.5.

Definition: Froehlich-Krumm distance (from [69])

Let $R = \langle r_1, \ldots, r_{|R|} \rangle$

Let $F = \langle f_1, \ldots, f_{|F|} \rangle$ where R, F are ordered sets of location point observations. Without loss of generality here they are considered as the prediction and the corresponding known truth respectively.

Let $LS(X) = \{s_1, \ldots, s_{(|X|-1)}\}$ be the set of line segments formed between each pair of consecutive points in the ordered set X, $X \in \{R, F\}$.

Let $Ind(s_r) = r$

Let $\delta(p, s)$ be the shortest Euclidean distance between point p and line segment l.

Then:

$$\textit{Froechlich-Krumm}(R, F) = \frac{1}{2}\left[\frac{\sum_{i=1}^{|F|} \min_{s \epsilon LS(R)} \delta(f_i, s)}{|F|} + \frac{\sum_{j=1}^{|R|} \min_{v \epsilon LS(F)} \delta(r_j, v)}{|R|}\right],$$

$s.t.$

$$\forall_{i>1} Ind\left[\arg\min_{s \epsilon LS(R)} \delta(f_{i-1}, s)\right] \leq Ind\left[\arg\min_{s \epsilon LS(R)} \delta(f_i, s)\right] \bigwedge$$

$$\forall_{j>1} Ind\left[\arg\min_{v \epsilon LS(F)} \delta(r_{i-1}, v)\right] \leq Ind\left[\arg\min_{v \epsilon LS(F)} \delta(r_i, v)\right] \quad (3.5)$$

In contrast to pointwise distance measures, the Froehlich-Krumm and Hausdorff distances seek to make a more relaxed comparison between trails automatically determining an the points/segments between which the measurements will be taken. Considering figure 3.2 it is clear that they are not perfect and that with good alignment in the samples the MAE can still perform in a way that is closer to what might be expected. In the next section the MAE and the Froehlich-Krumm distance is compared showing by example that, despite performing well when the trail observation points are well sampled, the MAE is inferior to the Froechlich-Krumm distance. Following this in section 3.1.7 a different distance measure is proposed which performs in a more intuitive fashion than

the Froechlich-Krumm distance.

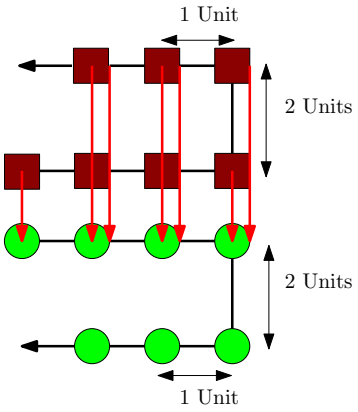
3.1.6 Froehlich-Krumm distance vs MAE

The Froehlich-Krumm distance and the MAE both seek to compute the average distance between a predicted route and the route that actually happened, but in slightly different ways. Importantly the Froehlich-Krumm distance automatically determines which parts in each trail to calculate distances between. In contrast the MAE uses the fixed order of the observation samples within each trail in a pairwise fashion. Specifically in this section a number of examples are considered showing that, in very common cases, performing pairwise comparisons is suboptimal. It also will be shown that this is true under uniform sampling through interpolation based on spatial distances. Recall that, as discussed at the beginning of section 3.1, it is not possible to interpolate based on time.

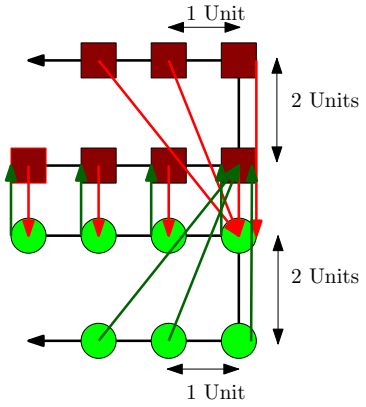
When considering the MAE and Froehlich-Krumm distance in reality it important to note that the trails may not contain the same number of point observations. In the case of the Froehlich-Krumm distance the default behaviour is somewhat intuitive in that the penalization in the case of extra length in one trail is based on the distance to the closest segment in the other trail. In the case of the MAE no default behaviour is present and so two cases are considered in all examples. The first approach is to simply compute sequential pairwise correspondences between points in the two trails, omitting any extra length in either trail. The second is to pair any extra points from one trail with the last point in the other trail.

An example where the differences in the distance measure discussed above matter is the the case of discretely sampled curves, where the prediction is made parallel, but on the inside of the curve. This is shown in figure 3.3. A common occurrence in real world scenarios, the problem with the MAE in this instance is that the alignment of the spatially uniform alignment (potentially provided by the recording device) of the points does not fit with basic intuition. Considering the example in more depth, however, it is possible to think of a fixed resampling technique that would provide a good alignment and that could be used as a pre-processing step to calculating the MAE. The resampling technique that would provide this is resampling the same number of points from each trail so that, in each trail, the points in each individual trail are evenly spaced spatially.

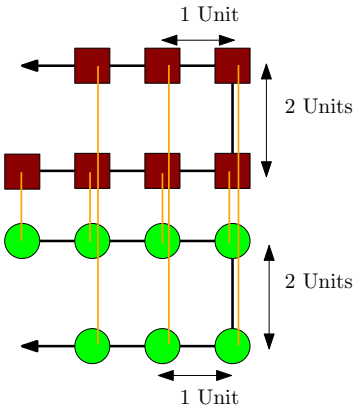
Considering the solution offered for the previous example a second example is shown in



(a) Distances considered by the Hausdorff distance.
hausdorff(,)



(b) Distances considered by the Froehlich-Krumm distance.
Froechlich-Krumm(,)



(c) Distances considered by the discrete Fréchet distance and MAE.
Fréchet(,)
MAE(,)

Figure 3.2: Exampling showing how different distance functions deal with the presence of ordered point observations within trails. (a) Hausdorff distance: Observation order is not taken into account (b) Froehlich-Krumm distance: Observation order is taken account when computing and then merging two asymmetrical distances (c) Fréchet distance and MAE: Observation order is taken into account at a global level. Note that MAE and Fréchet distances consider the same distances due to the specific sampling of the points.

figure 3.4. This time the two different ways of dealing with extra length are evaluated for three different sets of points. The first set are those provided originally, which includes a missing point and uneven sampling between the two trails. The second is the result of resampling uniformly in space. The third involves the aforementioned technique of sampling identical numbers of points. Despite these range of options, however, none provides an intuitive distance measure compared to the Froehlich-Krumm distance. This clearly highlights that, considering the two-dimensional nature of location within the route prediction problem, fixed pairwise alignment though sampling is not sufficient. Therefore distance measures such as MAE can not be recommended. Taking a closer look at the example in figure 3.4 it is also interesting that the problem noted in section 3.1.5 is again present, with the Froehlich-Krumm once more artificially under-weighting the distance incurred by the loop. A third example is provided in figure 3.5 which involves trails that cross over. The example once again highlights the inability of the fixed resampling schemes followed by pairwise point matching to successfully capture the differences between the curves in 2-D space.

In conclusion it is possible to see that (1) the MAE is not appropriate and (2) while performing closer to the general intuition, there are simple cases where the Froehlich-Krumm distance also does not perform correctly. In light of this an alternative metric is proposed and examined in the following section.

3.1.7 A solution: The average Fréchet Distance

In the previous sections the MAE and the Froehlich-Krumm distance measures have been examined. It was noted that the MAE was unable to adequately select the pairs of points to measure between trails, instead potentially relying on a pre-processing step of resampling. No resampling technique considered resulted in a satisfying result on simple examples. In contrast the Froehlich-Krumm distance suffered from a different problem. Based on the Hausdorff distance the measure suffers from the use of two averaged asymmetric distance calculations where, within each sub-calculation, not all points are required to match. As shown in figure 3.2 this can lead to a selection of point-to-point comparisons which underestimate the distance compared to what would be expected.

Due to the shortcomings of previously proposed/utilized distance measures for movement prediction, a modified version of the average discrete Fréchet distance is proposed here.

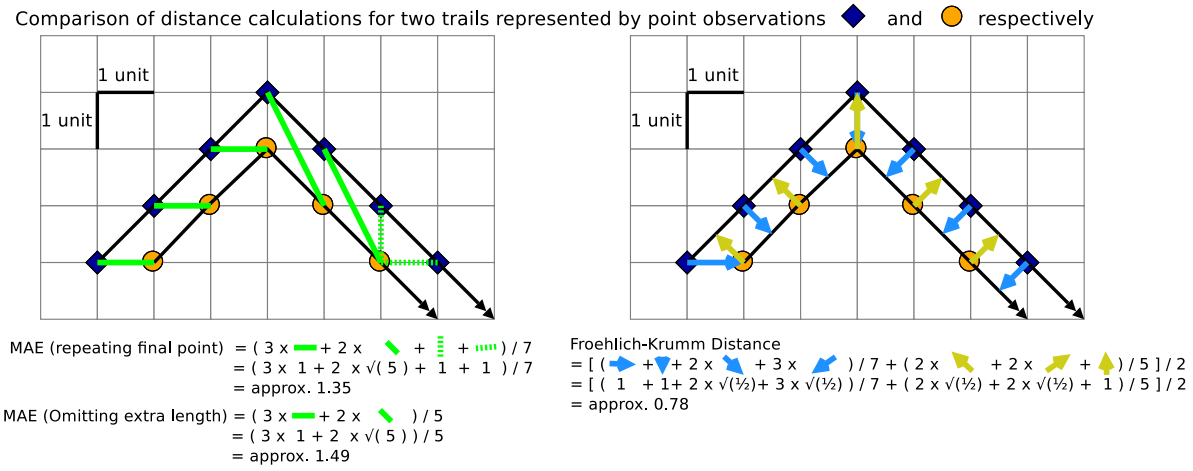


Figure 3.3: An example of the Froehlich-Krumm distance and the MAE when comparing between two paths represented by points diamonds and circles respectively. In this case the points in each trail have been recorded uniformly based on the distance travelled. For the MAE the distance is reported when the extra length is taken into account and when it is not. In the Froehlich-Krumm distance at each point in a, the closest point on the segments between all points in b is taken and the distance calculated and vice versa. In both cases the average distance is reported. Note that the MAE score is approximately 1.7-1.9 times larger than the Froehlich-Krumm distance, and that if the paths continued as indicated by the arrows, this difference would continue to increase.

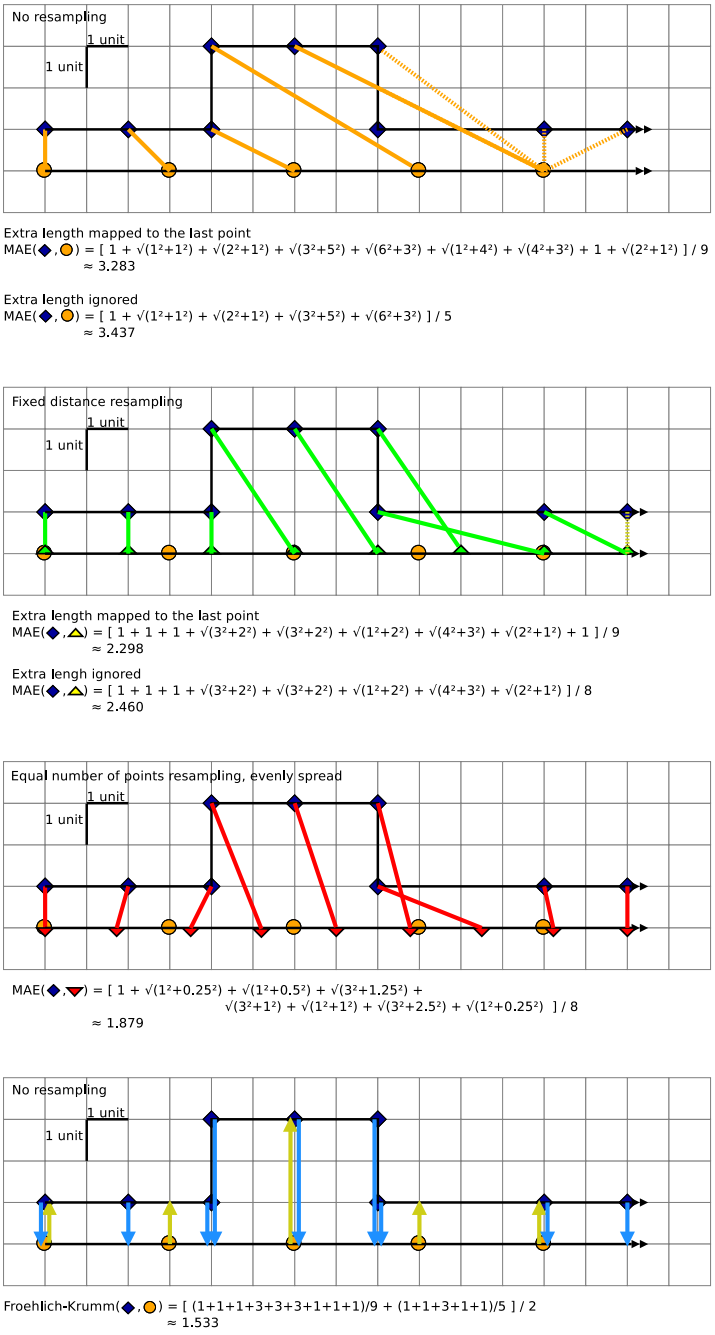


Figure 3.4: An example of the MAE and the Froehlich-Krumm distance when comparing between two trails represented by points shaped as diamonds and circles respectively. Evaluation of the MAE under three different resampling procedures and the two different approaches to dealing with extra length are shown. In the Froehlich-Krumm distance at each point in one trail, the closest point on the segments between all points in the other trail is taken and the distance calculated and vice versa. In all cases the average distance is reported. Note that the Froehlich-Krumm distance counts the *bump* relatively less, compared to the other sections which are counted twice before the weighted average is taken. As such in this case the Froehlich-Krumm distance underestimates the distance compared to what would typically be expected.

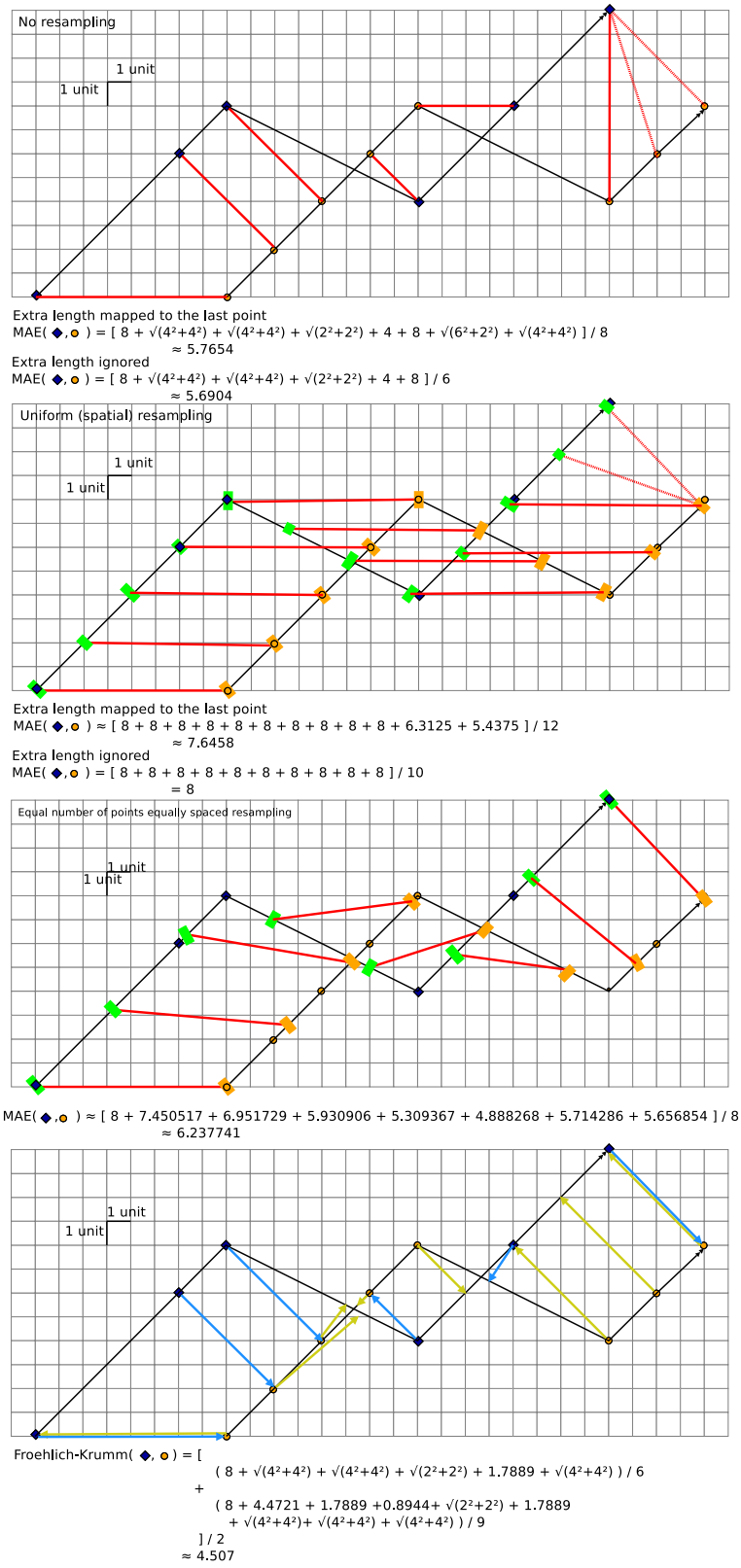


Figure 3.5: An example of the MAE and the Froehlich-Krumm distance when comparing between two trails that cross over. Each trail is represented by points shaped as diamonds and circles respectively. Evaluation of the MAE under three different resampling procedures and the two different approaches to dealing with extra length are shown. In the Froehlich-Krumm distance at each point in one trail, the closest point on the segments between all points in the other trail is taken and the distance calculated and vice versa.

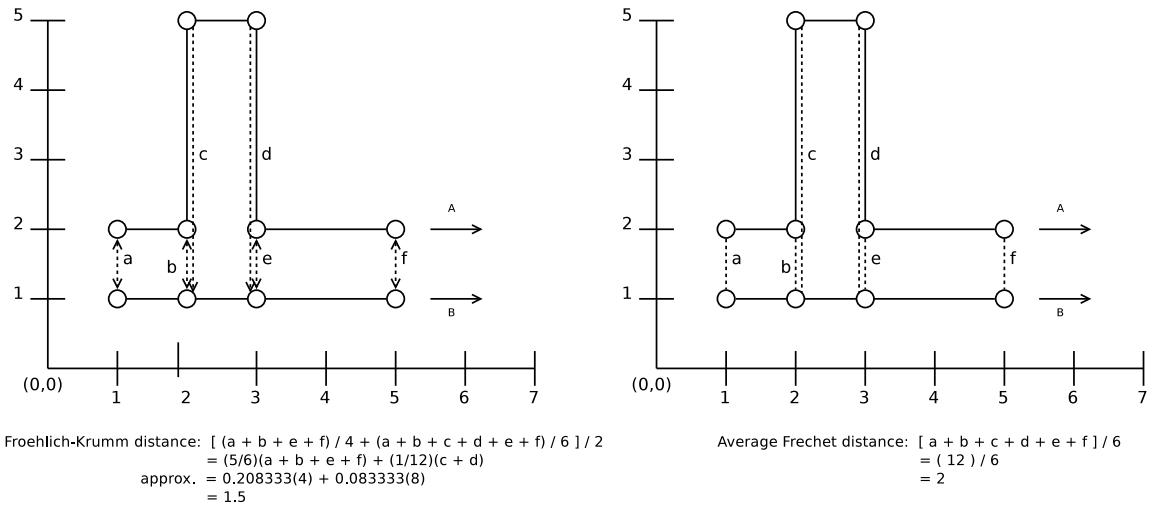


Figure 3.6: An example highlighting a case where the Froehlich-Krumm distance underestimates the distance compared to what would generally be expected, compared to the average discrete Fréchet distance where a more intuitive solution is achieved. Note, however, the effect of the discrete sampling at irregular intervals means that both measures do not give a perfect answer.

This modified version of the average discrete Fréchet distance is a variant of the discrete Fréchet distance [7] which has seen the application in numerous domains, including related domains such as map matching [28, 36, 183, 195]. It is of note that it is similar in spirit to the Froehlich-Krumm distance proposed by [69] and to the Hausdorff distance which the Froehlich-Krumm distance is based. In contrast to the Hausdorff distance the average discrete Fréchet distance takes into account the point observation order within the trails. With respect to the Froehlich-Krumm distance the average discrete Fréchet distance selects the point-to-point comparisons while simulationsly considering both trails and therefore does not suffer from the same problems observed earlier. As shown in figure 3.2, in contrast to the Froehlich-Krumm distance, the average discrete Fréchet distance automatically (in contrast to the MAE where this behaviour is fixed) considers the same set of point-to-point intuitive comparisons as the MAE. A further two examples are shown in figure 3.6 and 3.7 with the second the same example as in figure 3.5 showing crossed trails. Combined, figures 3.5 and 3.7 show an example where the automatic selection of the measured point-to-point comparisons is superior to the MAE under various resampling schemes.

It is of note that there also exists a continuous version of the Fréchet distance. This version is not utilized as it makes assumptions about the accuracy of all points sampled for each trail. Specifically the continuous version of the Fréchet distance integrates to get

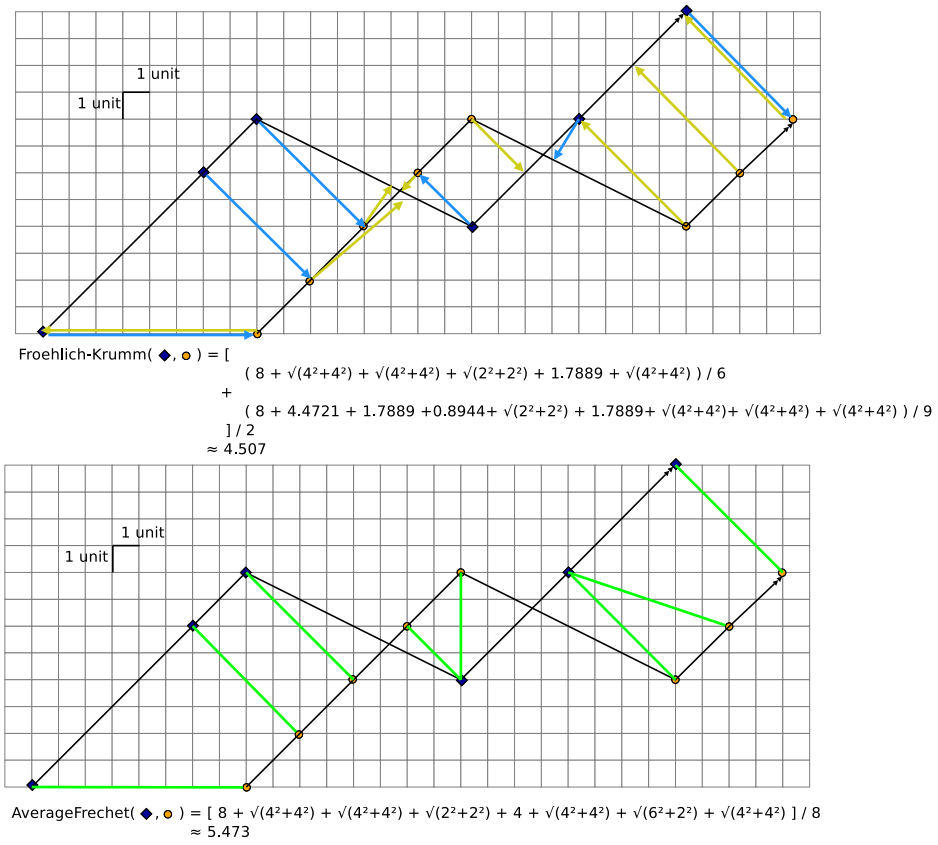


Figure 3.7: An example of the Froehlich-Krumm distance vs. the average discrete Fréchet distance for two trails that cross. An extension of the example in figure 3.5 the figure shows the more intuitive point-to-point comparisons selected by the average discrete Fréchet distance.

the area between the trails. Intuitively this can be seen as resampling via interpolation with an infinite resample rate. Since it is often the case that points *jump*, in other words a point is significantly in error, it is undesirable to interpolate all the way out to, and back from, that point adding point-to-point comparisons at each new interpolated point. Rather it is desirable that this point only contributes *one* sample to the overall error calculation. As such no resampling is preferred and the discrete version of the algorithm used. Further discussion will focus solely on the discrete version, with any discussion of the Fréchet distance referring to the discrete version.

When each trail is represented by a set of points the average discrete Fréchet distance can be interpreted as the average of the minimum total lengths connecting points across such that all points on the trails are occupied and the lengths do not cross over. Conceptually points from one path are matched to the closest point in the other under an ordering constraint. The ordering constraint used ensures that no backtracking along either trail occurs between inter-trail point pairs while allowing pauses of arbitrary length.

Formally the average Fréchet distance is defined as follows:
Let P, Q be polygonal curves made up of sequentially ordered points $R = (r_1, \ldots, r_{|R|})$ and $F = (f_1, .., f_{|F|})$ respectively. Once again, without loss of generality these curves will be considered the prediction result and its corresponding truth respectively.
Define a *coupling* L between R and F as a sequence of point pairs where each pair comprises of one point from curve R and one point from curve F such that all points from R and F are used at least once:

$$L = \langle (r_{a_1}, f_{b_1}), (r_{a_2}, f_{b_2}), \ldots, (r_{a_m}, f_{b_m}) \rangle$$

In each pair a and b refer to indexes into P and Q respectively. Note that in each pair a and b may take different values. The additional subscript on the indexes denote the order of the point represented by the index in the coupling L. A sequence is only a

coupling if it satisfies the constraints:

$$a_1 = 1$$

$$b_1 = 1$$

$$a_m = |R|$$

$$b_m = |F|$$

$$\forall_{i=1,\ldots,q}(a_{i+1} = a_i \vee a_{i+1} = a_i + 1) \wedge (b_{i+1} = b_i \vee b_{i+1} = b_i + 1)$$

The constraints $a_1 = 1, b_1 = 1, a_m = |R|, b_m = |F|$ ensure that the first pair in the sequence is the pair of first points in the curves and that the last pair is the end points from both curves. The remainder of the constraints force a coupling to respect the sequential order of the points in R and F. A coupling denotes a potential set of pairwise points between R and F. There are a large number of such couplings. Let $\mathcal{L}$ be the set of all possible couplings.

The discrete sum Fréchet distance [60] is then the coupling that minimizes the following function, where $d(\cdot, \cdot)$ is an arbitrary[1] distance function between two point observations:

$$\mathit{FréchetSum} = \min_{L \in \mathcal{L}} \sum_{i=1}^{m} d(r_{a_i}, f_{b_i}) \tag{3.6}$$

The average discrete Fréchet distance is then defined as:

$$\mathit{FréchetAvg} = \frac{\min_{L \in \mathcal{L}} \sum_{i=1}^{m} d(r_{a_i}, f_{b_i})}{m} \tag{3.7}$$

An example showing two potential couplings (including the selected minimal coupling) as considered by the average discrete Fréchet distance is shown in figure 3.8.

The average discrete Fréchet distance can be calculated in the **R** statistical computing software [158] with some small modifications to the *pathFrechet* function in the package *longitudinalData* [72][2]. The modifications are provided as a new function in appendix A.

While the average discrete Fréchet distance has been motivated as being able to find good point-to-point comparisons to measure it is still restricted to choosing between the

[1] In this work the Euclidean distance function is used.

[2] Which in turn is based on the algorithm from [60]

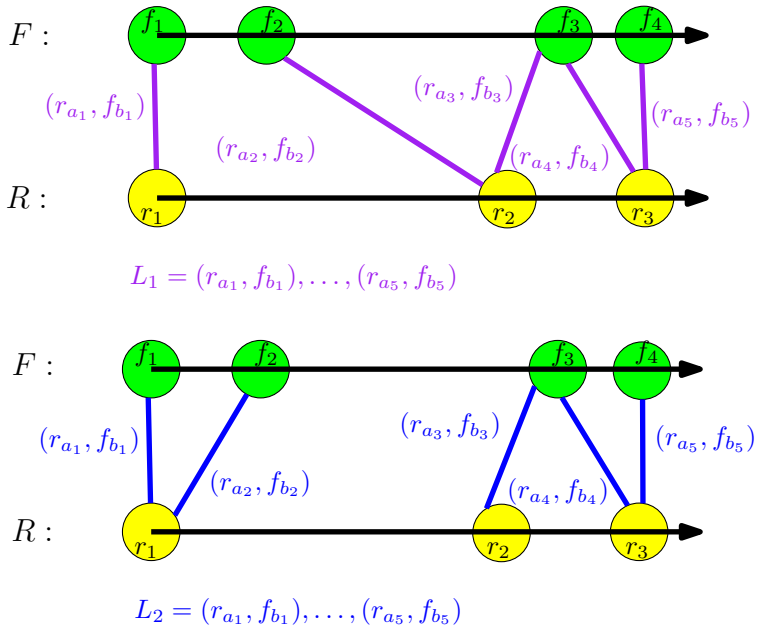


Figure 3.8: A example of two trails, R and F of different lengths, showing two potential couplings (L_1 and L_2). In this case L_2 is the coupling with the minimal score by equation 3.7 and its value by this equation is the average Fréchet distance. Coupling L_1 represents another valid coupling, but not the minimal coupling.

sample points available. As previously discussed it is undesired to introduce interpolated sample points due to the nature of GPS and the noise it contains. Unfortunately, however, the other form of error within GPS, missing symbols means that it would be unwise to not perform *any* interpolation. This is easily illustrated in figure 3.9 as an extreme case where a large number of points from the second trail as missing. As is easily visible, this forces the discrete Fréchet distance to measure undesirable point-to-point distances. Therefore, while undesirable in general, some resampling needs to be performed. In this thesis linear interpolation is used to resample and the smallest segment length in the two paths being compared used as the segment length. The ability to perform this resampling automatically is also provided as part of the new function in appendix A.

A final concern is differing lengths between the prediction and ground truth. Here, two cases must be considered, (1) the prediction is shorter than the ground truth and (2) the prediction is longer than the ground truth. In the former it would generally be expected that the shorter prediction would get penalized for making a shorter prediction, in the latter the case is not immediately as clear. On one hand it may be desirable to predict the end point and hence longer predictions should be penalized. On the other hand it could be argued that as long as the prediction followed the ground truth then it was correct,

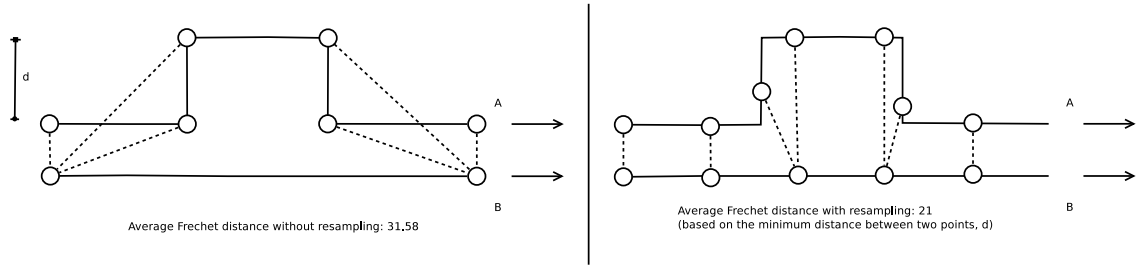


Figure 3.9: An example highlighting the need for resampling in the presence of missing samples.

regardless of what it did afterwards. The discrete average Fréchet distance addresses the case of short predictions by penalizing the prediction at a cost of the distance between the final point on the prediction and the point further along in the ground truth sequence before the average is taken. This seems reasonable. In the case of long predictions the same penalization is applied, although in this case the penalty is occurred for each additional point in the prediction. Unfortunately this can lead to the unexpected result of *reducing* the overall average score if, for instance, the extra length stays very close to the final point in the ground truth. As such either a strict penalty must be applied or this possibility considered acceptable. Either case is suboptimal for measuring performance since the exact value of the penalty that should be applied is unclear. Due to this, in this thesis any extra prediction length is not considered. Both cases can be implemented as a modification of the average Fréchet distance. This is achieved by examining the selected minimal coupling L_{min} (see equation 3.8) based on the summed Fréchet distance penalty from which average Fréchet distance is calculated

$$L_{min} = \underset{L \in \mathcal{L}}{\operatorname{argmin}} \sum_{i=1}^{m} d(r_{a_i}, f_{b_i}) \tag{3.8}$$

The truncated average discrete Fréchet distance which takes into account the desire *not* to penalize extra prediction length is therefore defined here as:

Let $R =$ the prediction curve, with a unaltered definition otherwise

Let $F =$ the curve of what actually happened, with an unaltered definition otherwise

Recall: $R = \langle r_1, \ldots, r_{|R|} \rangle$ and $F = \langle f_1, .., f_{|F|} \rangle$

Recall: L is an arbitrary coupling of (R, F), $L = \langle (r_{a_1}, f_{b_1}), (r_{a_2}, f_{b_2}), \ldots, (r_{a_m}, f_{b_m}) \rangle$

Recall: $d(\cdot, \cdot)$ is a arbitrary distance function between two point observations

Recall: $\mathcal{L}$ is the set of all possible couplings of (R, F)

Let $FI(L)$ return the first index i of a pair (r_a, f_b) in L in which $f_b = f_{|F|}$

Let $L_{min} = \underset{L \in \mathcal{L}}{\operatorname{argmin}} \sum_{i=1}^{m} d(r_{a_i}, f_{b_i})$

$$\textit{FréchetAvgTrunc(R,F)} = \frac{\sum_{i=1}^{FI(L_{min})} d(r_{a_i}, f_{b_i})}{FI(L_{min})} \tag{3.9}$$

In other words, the summation of L_{min} is stopped once the last point in the ground truth trail has been accounted for. At this point it is possible to either use the sum as is (as per equation 3.9) or to add a custom penalty for the additional length in the prediction, in both cases adjusting the divisor accordingly. The truncated average discrete Fréchet distance is implemented with an flag (default to FALSE, as used in this thesis) to indicate whether extra distance is to be penalized. The code additionally resamples based on the smallest segment by default as previously discussed with a flag to disable this behaviour. Custom penalties for extra length are currently not supported although this modification is quite straightforward. The R code is provided in appendix A.

3.1.8 Aggregating prediction scores

In the previous sections distance measures used to measure the prediction quality of a single prediction were discussed. The goal, however, is to measure the prediction quality of individual predictors not just the prediction quality for a single prediction by a specific predictor. In order to achieve this goal it is important to consider the performance of the predictors across multiple predictions. This requires a method for aggregating a set of prediction results. A decision on this method is required before discussing the end goal of a more general measure between two predictors. In line with later use the aggregation considered here is the overall prediction quality calculated across a number

of predictions given a trained predictor (in other words for a fixed training set $\mathcal{K}$).

This aggregation is calculated by creating a test set of full historic trails $\mathcal{T}$, that are (generally) not part of $\mathcal{K}$, and segmenting them to create a set of (input, observed result) pairs as required for individual trail prediction. Specifically $\forall T \in \mathcal{T}, T = \langle E, F \rangle$ where $E = \langle e_1, \ldots, e_{|E|} \rangle$, $F = \langle f_{|E|+1}, \ldots, f_{|T|} \rangle$ and e and f represent point observations (feature vectors) including location (e.g. [longitude, latitude]). For convenience a predictor is formalized as a function $\mathcal{P}(\mathcal{K}, \cdot)$ that takes as arguments a training set $\mathcal{K}$ and a arbitrary observation (indicated by the $\cdot$) from which a prediction is made. Given a set of test cases $\mathcal{T}$ a naive approach would be to simply take the mean, a common measure of central tendency, over all predictions. Taking this approach the aggregation of prediction error (defined as PE) of predictor $\mathcal{P}$, given training set $\mathcal{K}$ and test set $\mathcal{T}$, is:

$$PE(\,\mathcal{P}(\mathcal{K}, \cdot), \mathcal{T}\,) = \frac{\sum_{i=1}^{|\mathcal{T}|} \mathit{FréchetAvgTrunc}(\mathcal{P}(\mathcal{K}, E_i), F_i)}{|\mathcal{T}|} \tag{3.10}$$

Unfortunately this does not turn out to be an aggregation that reflects the overall quality of the predictor in practice. Specifically it does not provide a good measure of central tendency because the distribution of individual prediction errors are not normally distributed. To illustrate this prediction error histograms from predictors evaluated in Chapter 6 are shown in figure 3.1.8. In each case the histogram shows results for a predictor (for more details on the predictor see chapter 6) which has been trained on a fixed historic set ($\frac{9}{10}$ of the whole data set) and evaluated on a held out test set ($\frac{1}{10}$ of the whole data set). The result is that the mean is taken in the direction of the skew. Since the error can never be less than zero, this always increases the mean compared to what might be expected. In addition the mean can be quite susceptible to outliers.

For example consider two arbitrary predictors, A and B, making twenty different predictions. Consider the case where the MAE is used to evaluate the predictors and that the MAE value can range from zero (best) and an arbitrary large upper bound (worst). If method A makes 10 exactly correct predictions (zero MAE) but also 10 completely wrong predictions (MAE value of 50) then the method would average 25. In contrast say method B makes 20 predictions with a MAE value of 20, resulting in an average of 20. In this case, via averages, method B would be considered superior. However, this is an undesirable conclusion if we consider that the MAE values could have been referring to the average distance at each time step between the prediction and correct result in *metres*. As such method B always makes wildly incorrect predictions (on average out

by 20m at each point) whereas method A makes correct predictions 50% of the time and wildly incorrect predictions the other 50% of the time, making it a more useful prediction method that B.

An alternative measure of central tendency that is generally recommended for skewed data is the median. The median represents the middle value of the set of prediction errors and is the value that separates the upper half from the lower half. With regard to route prediction the median indicates, for a given predictor for that test set, that 50% of the predictions were better than that value.

Here, however, a third option is proposed. This is the use of a binary decision function based on a user-defined level of an acceptable prediction at the level of an individual prediction. The aggregation (via the mean) can then be interpreted as the proportion of predictions which were acceptable. This approach makes use of the fact that in many applications predictions are only useful to a certain threshold. This results in a proportion similar to evaluation approaches that rely on exact matching. However, the approach allows an arbitrary specification of the accuracy for which a prediction is deemed correct, decoupling this from the level of symbolic quantization. In addition the approach does not exclude the examination of the predictions distance values before applying a threshold.

When deciding between the median and the proportion approach it is important to realise that just because predictor A has a lower median than predictor B does not mean that predictor A makes more predictions below a certain threshold than predictor B. Real world examples where the median for predictor A is smaller than the median of predictor B while predictor B has predicted more below a relevance threshold of $5m$ are shown in figure 3.1.8. In the figure the first column represents predictor A and the second predictor B. Each row then represents a comparison of interest. Considering pairwise comparisons between all test/training sets over all predictors used in chapter 6 these cases occur in 17.29% of the pairs.

While either could be used they therefore provide different views on predictors performance and this may lead to different conclusions as to which predictor is *better*. Here the proportion based approach is considered superior as, after selecting a application specific threshold, the results clearly define how fit for purpose the various prediction algorithms are. Specifically the proportion of acceptable predictions on average the predictor will make is measured. In addition it is important to note that a general understanding of

the performance of the predictors over varying thresholds can still be obtained via descriptive statistics and visualizations. In fact this is recommended later in this chapter with a number of visualizations provided for this purpose in section 3.3

In contrast the median only provides a notion that at least 50% will be better than the value reported in the evaluation. While the median will provide a (potentially different) indication of which predictor is *better* it does not provide a concrete indication of whether an algorithm is fit for the purpose of the target application. Of course the answer could be to always select the best performing predictor, as defined by the lowest median. However, as noted, this still may not make the most correct predictions to the level deemed acceptable by the application. Additionaly, as discussed in more depth in chapter 4 the best predictor may not be the one desired, rather the desired predictor may be one that is computationally efficent while still maintaing good prediction quailty within the application specific threshold.

Based on the above discussion the following definition of prediction quality of an individual predictor ($\mathcal{P}$) for a given training set ($\mathcal{K}$), test set ($\mathcal{T}$) and user-defined level of an acceptable prediction (α) is advocated and used throughout this thesis:

$$PE(\,\mathcal{P}(\mathcal{K},\cdot),\mathcal{T}\,) = \frac{\sum_{i=1}^{|\mathcal{T}|}\gamma(\,\mathit{FréchetAvgTrunc}(\mathcal{P}(\mathcal{K},E_i),F_i)\,)}{|\mathcal{T}|}$$

where:

$$\gamma(x) = \begin{cases} 1 & \text{if } x < \alpha \\ 0 & \text{otherwise} \end{cases} \tag{3.11}$$

3.1.9 Summary: A distance metric for movement prediction

Measuring the quality of a prediction in a way that is meaningful in practice is an important issue. Much previous research has focused on the use of proportions through binary similarity functions based on the underlying symbolic encoding. Rarely, however, is the effect of the previously chosen level of symbol quantization on evaluation considered. One solution is to use a distance function which then provides a more detailed understanding of the difference between the prediction and the correct answer that can be achieved. With a straightforward interpretation the mean average error (MAE) provides an intuitive and widely used measure of the difference between two time series.

However, even if resampling is used to align the samples, the results do not reflect the general intuition. To this end [69] proposed a Hausdorff based distance metric. While performing better, it still has shortcomings and so the truncated average discrete Fréchet distance was proposed. Having defined the prediction error for an individual prediction the problem of aggregating the prediction error of a trained predictor (PE) was discussed. It was noted that the error distribution of the predictions is not normal but rather positively skewed. In addition it was noted that outliers have the potential to cause serious issues. To address this an approach that results in proportions derived from a user-defined threshold defining a useful prediction is advocated. The approach addresses the issue by classifying each individual prediction as either useful or not based on user-defined level of an acceptable prediction. The threshold is based on the truncated average discrete Fréchet distance providing a threshold with a well defined and intuitive interpretation. Finally graphs of the proportions over varying thresholds is advocated providing a simple yet informative view of comparative performance. These are detailed in section 3.3. Statistical testing, however, must be carried out at the specific levels of interest. This is discussed further in section 3.2 along with how to calculate the actual prediction error of interest, the prediction error of a predictor independent of the specific training set.

3.2 Testing methodology: Comparing predictors

Evaluation of the performance of different prediction algorithms is clearly an important task. Here the goal is to evaluate the question *is there a difference in prediction performance between two predictors?*. Performance here is defined as the prediction quality of a predictor independent of the training set and prediction instance. This can be defined as the expected[3] performance of predictors trained with an arbitrary training set. Calculating such a statistic involves varying not only the test data as done to calculate the PE but also varying the training set used. Additionally it is important to note that the predictor error is also dependent on the size of the historic training set used. This parameter is generally not varied, however, as it is expected that learning algorithms will not decrease their performance as the number of training instances is increased. Additionally when used in practice the amount of training data is expected to increase with time. Therefore it does not make sense to evaluate the performance of predictors

[3]The expected performance is the mean performance.

with less than the maximum possible training set size. In the case of route prediction these arguments certainly seems reasonable since historic observations are continuously being generated by the system.

Formally the expected prediction performance, EPE, for a predictor $\mathcal{P}$ trained on an arbitrary training set $\mathcal{K}$ of fixed size $|\mathcal{K}|$ can be defined as:

$$EPE_{|\mathcal{K}|}(\mathcal{P}) = E[PE(\mathcal{K}, \cdot), \mathcal{T})] \tag{3.12}$$

Where $E[\cdot]$ denotes the expectation (mean) and T denotes arbitrary test sets. The function $PE(\cdot, \cdot)$ is as defined in equation 3.11.

Having defined the measure of interest it is important to recognize that each predictor, $\mathcal{P}$, has a true $EPE_{|\mathcal{K}|}(\mathcal{P})$ value. However, without enumerating the whole population of trails it is not possible to calculate this value. Therefore this must be estimated. Since it is an estimated quantity it also important to assess the uncertainty surrounding this estimation. In assessing the uncertainty the distribution of the sampled PE values is typically assumed to be normal. By making such an assumption and calculating the variance of the EPE (denoted as $Var[EPE]$) standard statistical tests such as paired t-test or paired difference test can be applied to determine if statistically significant differences exist between predictors. Formally these tests answer the question of *is there a difference in prediction performance between two predictors A and B?* through the null hypothesis:

$$H_0 : EPE(A) = EPE(B) \tag{3.13}$$

The general form of the estimated EPE given a finite amount of data is:

Recall $\mathcal{K}$ denotes a training set of historic trail observations

Recall $\mathcal{T}$ denotes a test set of $\langle E_i, F_i \rangle$ pairs

Let $\mathcal{Q} = \{Q_1, \ldots, Q_J\}$ be a set pairs of test and training set pairs, $Q_j = \langle \mathcal{K}_j, \mathcal{T}_j \rangle$

Let J be the number of PE values that go into the estimate of the EPE, $J = |\mathcal{Q}|$

Recall $\mathcal{P}$ denotes a predictor

Recall $PE(\cdot, \cdot)$ is the prediction error for a trained predictor over the given test set defined in equation 3.11

Then:

$$EPE_{|\mathcal{K}|}(\mathcal{P}, \mathcal{Q}) = \frac{\sum_{i=1}^{J} PE(\mathcal{P}(\mathcal{K}_i, \cdot), \mathcal{T}_i)}{J} \tag{3.14}$$

As will be evident in section 3.2.3 the approaches to estimating the $Var[EPE]$ do not have a general form.

The assumption that the PE samples are normally distributed is defended by highlighting that each PE value is the average of a set of predictions from a predictor using a sampled training/test set pair. Therefore the distribution in question is *not* the Bernoulli distribution of individual prediction outcomes, nor is it previously discussed (section 3.1.8), non-normal, error distribution of the truncated average discrete Fréchet distance. Rather, it is the distribution of sample sample proportions. This distribution can be argued to be approximately normal using the the central limit theorem. With respect to proportions the central limit theorem states that as the sampling size increases the sampling distribution of the proportion more closely approximates the normal distribution (see, for example, [125]). Therefore in the rest of the discussion here the distribution of PE values will be assumed to be normal.

In practice, however, it is still important to check the assumption of normality, particularly if the sample from which the proportions are derived is not very large. Numerous tests for normality exist, many of which are implemented in popular statistical packages. Throughout this thesis the D'Agostino normality test is used which has shown good performance compared to other methods [49, 156]. This test is readily available in the **R** statistical computing software [158] in the form of the function *dagoTest* in the *fBasics* package [199]. In addition to the tests graphical plots are typically examined.

Recommended [49] is the normal probability plot in which the data is plotted against a theoretical normal distribution, represented as a straight line. Deviations from this line indicate non-normality. These graphs can easily be obtained in popular statistical packages. In **R** [158] this can be achieved using the *qqnorm* and *qqline* commands.

Returning to the issue that the EPE (the expected value of the PE for different training sets) and $Var[EPE]$ can not be exactly calculated and must be estimated the goal becomes to provide a *good* estimate of the EPE and additionally a *good* estimator of the $Var[EPE]$. A good estimator is defined by considering that each estimator (of the EPE or $Var[EPE]$) has its own distribution (distinct from the distribution of PE values). If the mean of this distribution equals the true EPE or $Var[EPE]$ then the estimator is considered good and defined to have zero *bias*. If an estimator does not have a zero bias then it is said to be biased and is not considered to be a good estimator. A large amount of work has been done in mathematically proving estimators are either biased or unbiased and, if they are, whether the estimator over- or under-estimates the EPE or $Var[EPE]$. This is important as, for instance, over-estimating the variance generally leads to liberal statistical inference. In other words the tests report differences *that do not exist in reality*. It is this body of work that is critically discussed herein with respect to route prediction evaluation and a testing methodology specific to route prediction synthesised. This is important to ensure that any results reported are correct and have not been over- or under-estimated through the choice of a specific testing methodology.

3.2.1 Basic approaches to estimating the mean and variance

Let the full data set of historic trails be denoted by D. It is important to note that if the data set is large enough a unbiased EPE and $Var[EPE]$ can be easily found. Recall that $\mathcal{K}$ denotes the historic trails used to train a predictor and $\mathcal{T}$ denotes the set of historic trails that form the test set. For a fixed training set size $|\mathcal{K}|$ and a large test set size $|\mathcal{T}|$, this involves randomly sampling $|\mathcal{K}|$ trails from D to form $\mathcal{K}$ and a further $|\mathcal{T}|$ trails from D to form $\mathcal{T}$. $\mathcal{K}$ and $\mathcal{T}$ then form a training/test pair $Q_j = \langle \mathcal{K}_j, \mathcal{T}_j \rangle$. With a large enough data set this can be repeated J times resulting in J sampled training/test pairs, $\mathcal{Q} = Q_1, \ldots, Q_J$. Large enough here refers to the ability to effectively, repeatedly, randomly sample independent test and training set which can also be considered independent across samples. For each $Q \in \mathcal{Q}$ an independent estimate

of the prediction error (PE) can be made via equation 3.11 leading to J such estimates. From these estimates an estimate of the EPE and the $Var[EPE]$ with respect to the specific size of the training set used can be obtained using standard formulas. In practice, however, the data set is of finite, often small, size and all estimates must be made from this one data set. Since the data set is small simple resampling leads to dependant training/tests sets with the dependency based on the small selection of trails in the data set compared to what exists in reality.

Other basic solutions such as training and testing on all data ($\mathcal{K} = D, \mathcal{T} = D$) or simply partitioning the data into disjoint training and evaluation sets have long been known to be flawed approaches in the case of limited data. In both cases only one training and one test set is used ($J = 1$) and so only the prediction error of the trained predictor is determined. In addition, the former results in overfitting where the results fail to generalize to unseen data. The latter does not use all data and the results are highly dependent on how the disjoint sets are formed, reducing the replicability of the tests over the same dataset and algorithms. This has led to a large body of work with respect to resampling methods which seek to reuse the data in a systematic and principled way. Specifically these methods focus on the generation of $\mathcal{Q}$, the set of sample training/test set pairs. In order to adequately discuss the issues with respect to systematic data reuse discussion is divided into two, one considering the requirement for an unbiased estimator of the EPE and the other consider the requirement for an unbiased estimator of the $Var[EPE]$.

3.2.2 Estimators of mean error under systematic data reuse

The most common methods for systematic data reuse that have seen in-depth investigation [25, 27, 106, 107, 136] are variations of cross validation and bootstrap procedures.

In k-fold cross validation the set of training and test set pairs, $\mathcal{Q} = Q_1, \ldots, Q_J$, is formed by constructing k disjoint sets by partitioning the data into k groups of equal size. Each group is held out once as the test set and all others used as the training set, arriving at k training/test set pairs ($J = k$). The value of k therefore controls the size of the training set. Note that k-fold cross validation is an unbiased estimator of the $EPE_{\frac{k-1}{k}|D|}$. In other words the estimator is unbiased for the EPE of a predictor with a fixed training set size of $\frac{k-1}{k}|D|$ [141]. However, as previously discussed it is

desired to evaluate the predictors for the largest possible training set size as in reality the full data set would be used as training data. A larger training set can be obtained by increasing the number of folds. In the extreme, leave-one-out cross validation produces an estimate of the EPE for a training set size close to that of the data set size. This comes at the cost of high variability. Variability here refers to the degree of difference in the reported EPE values depending on the randomisation used in the sampling of the training/test sets. As a trade off between training set size and variability, ten folds is recommended in [107] and has become commonly used in practice. More recently, however, more computationally intensive methods are recommended, such as repeated k-fold cross validation or bootstrap approaches [25, 161].

An extension to k-fold cross validation, repeated k-fold cross validation involves repeating the k-fold cross validation procedure an arbitrary number of times. Before each repeat the data set is permuted and the partitioning applied based on the order of the data. This results in different combinations of the individual historic instances being part of each group. The goal here is to eliminate effects of a specific data partitioning of a single k-fold cross validation run and provides additional estimates of the prediction error which can then be averaged.

In contrast to cross validation, bootstrapping [58] generates a set of samples for training equal to the size of the data set, $|\mathcal{D}|$, by sampling randomly with replacement. Since sampling is done with replacement not all historic trails from the data set will appear in the training set. Specifically the probability that a given historic trail will not be selected is 0.368. The historic trails not selected then form the test set. This leads to a test set of approximately $0.368|\mathcal{D}|$ and a training set of approximately $0.632|\mathcal{D}|$. Repeating this procedure k estimates can be obtained and averaged to achieve an error estimate err_1. Compared to 10-fold cross validation, for instance, this training set is generally made up of a significantly smaller number of unique observations which typically leads to an overestimation of the prediction error. The .632 and .632+ bootstrap methods aim to correct for this by combining this error with an error known to underestimate the prediction error, the resubstitution error, err_2. The resubstitution error is the prediction error when the training set is used for both training and testing. The .632 and .632+ bootstrap methods are then of the following form:

$$err_3 = \beta err_1 + (1 - \beta) err_2$$

where $\beta = 0.632$ for .632 bootstrap and β is based on the *no-information error rate*

[59] in the case of .632+ bootstrap. The bootstrap .632+ and others are not discussed in more depth as in general the bootstrap procedure can suffer some complications in the presence of perfect memorizers[4]. This can be clearly seen when considering that perfect memorizer will always return a zero resubstitution error. This problem also holds for the correction applied to the corrected repeated k-fold cross-validation from [32] since the adjustment involves prediction error calculations where the same historic instances are in both the test and training sets. Since a number of predictors within the movement prediction approaches tend toward perfect memorizers (e.g. the pattern matching algorithm in [137]) these methods are considered less appropriate. Of the remaining resampling methods, repeated k-fold cross validation is generally recommended [25, 106], with the generation of training/test sets via repeated random sampling without replacement generally shown to provide more pessimistic estimates of the error.

In light of the above discussion repeated k-fold cross validation is recommended. Throughout this thesis k is set to 10 following the recommendation in [107] and the number of repetitions is set at 10 based on computational considerations.

3.2.3 Estimators of variance under systematic data reuse

When re-using the available data to estimate the EPE the variance, as would normally be calculated, is affected. This is because the true (population) variance is normally estimated via the sample variance, since the true variance can not be calculated. This calculation involves the assumption that the samples (in this case the PE) are independent. When data resampling techniques are used to generate the PE samples the assumption is violated leading to a potentially large underestimate of the variance [16, 26, 141]. This is problematic as it can result in incorrect conclusions of significant differences in cases where there are none in reality (Type I error[5]). This is prevalent under most standard tests such as the t-test. The reason that the PE values are not independent is as follows. Recall that the EPE is the mean of J estimates of the PE, each calculated from a given training ($\mathcal{K}$) and test ($\mathcal{T}$) set pair (Q). Since the training/test set pairs are not independent then neither are the J PE estimates.

Unfortunately it has been shown that no unbiased estimate of the variance exists for k-

[4]Perfect memorizers are predictors which remember all training data and can therefore always predict observations repeated in the test and training set correctly.

[5]Formally Type I errors are errors in which the null hypothesis is rejected when it is in fact true.

fold cross validation [16], and currently no unbiased estimates exist for other resampling techniques [141]. In light of this a number of authors have proposed solutions that either aim to provide estimates with small bias ([131, 141]) or that are guaranteed to overestimate the variance and hence guarantee to only show significant results when they exist at the expense of failing to report some cases where significant cases did exist in reality (Type II errors[6]) [141].

The rest of the discussion on variance is divided into two sections that individually discuss attempts to provide estimates with small biases and overestimate the variance respectively. A third a final section concludes the discussion. Throughout it is important to remember that the variance being determined is the variance of the estimated EPE calculated using one of the aforementioned techniques for splitting and re-using the data set, D. For example the variance being calculated in the case of k-fold cross validation is $Var\left[EPE_{|\mathcal{K}|}^{|\mathcal{T}|}\right]$ where $|\mathcal{K}|$ is the size of the training data set ($(|\mathcal{K}| = 1 - \frac{1}{k}) \times |D|$) and $|\mathcal{T}|$ is the size of the test set ($|\mathcal{T}| = |D| - |\mathcal{K}|$).

Variance estimates with small bias

Providing an in-depth analysis [141][7] consider the problem of variance estimation and statistical inference, first proposing a estimate of the $Var[EPE]$ with a small bias and then evaluating it embedded in resampled t-tests and bootstrap inference.

The proposed estimator acknowledges that it is the correlation between the sampled means (the PE values) that is unknown (and ignored when naively calculating the variance). The estimator is based on the assumption that a prediction outcome is only dependent on the test trail and the size of the of the training set. In other words, that the prediction is not dependent on the exact trails used to train the predictor, but only on the size of the training set used. When the training size is very large it is possible that this is true. The authors then show that under this assumption the correlation among the PE's is then equal to the size of the training set divided by the sum of the training and test set sizes $\left(\frac{|\mathcal{T}|}{|\mathcal{K}|+|\mathcal{T}|}\right)$. Assuming this formulation for the covariance between the

[6]Formally Type II errors are errors in which the null hypothesis is not rejected when it is in fact false.

[7][141] refers to the 2003 journal publication, however, this was work written significantly earlier in 1999 as a working paper [17].

PE's the following estimate of the $Var[EPE]$ is constructed:

$$Var\left[EPE\right] = \left(\frac{1}{J} + \frac{|\mathcal{T}|}{|\mathcal{K}|}\right) S^2_{PE_j} \tag{3.15}$$

where $S^2_{PE_j}$ is the sample variance of the J PE's, calculated via the standard formula:

$$S^2_{PE_j} = \frac{1}{J-1}\sum_{j=1}^{J}(PE_j - EPE)^2 \tag{3.16}$$

As discussed and shown empirically by the authors, the $Var[EPE]$ estimate is then biased by an undetermined amount depending on how well the assumption holds. In any case, however, it is guaranteed to provide a better estimate of the $Var[EPE]$ as simply using the sample variance directly assumes a correlation of zero between the PE's. However, by not considering the dependence introduced by the specific training observations it is possible that the variance is still underestimated since the correlation between the PE's as calculated by $\left(\frac{|\mathcal{T}|}{|\mathcal{K}|+|\mathcal{T}|}\right)$ will be underestimated.

As noted by the authors, this means that the approximation is good when the algorithm is somewhat agnostic to the exact training data used, for the given training set size. In other words, if changing the exact observations in the training set does not change the prediction performance much. This means that the decision of whether to use the approximation should be done considering the predictors being tested. In practice it is hard to know if, for a given training data size, the algorithms change their performance significantly when altering the observations in the training set. At first glance it may seem that a Markov model might be sensitive to alterations in the training data. However, if the training set is large and the overall data set has a large number of repeated trails then it may not be overly sensitive. This is investigated further empirically later in section 3.2.4.

[131] continue the same line of work proposing another variance estimator, based on a more restrictive set of assumptions, primarily on the type of the prediction error function used. The authors analysis focuses on differentiable prediction error functions, for which they show their method to have lower bias and less variance compared to the variance estimate previously discussed from [141] (equation 3.15) under the same cross-validation sampling. While low, however, the bias reported is always negative, in other words a underestimate of the variance, the result of which may be liberal inference, although this is not examined. Unfortunately the assumptions that the prediction error function be differentiable is false in the case of the prediction error function motivated in this thesis as

it is an indicator function[8], which is discontinuous at one point. Additionally occurring in classification [131] provide preliminary results using an polynomial approximation to the the indicator function. However, the authors note that further research is required before this approach can be recommended. As such this method should not be used when evaluating movement prediction algorithms.

In [26] the issue of hypothesis testing between two learning algorithms is explicitly examined in the classifier setting. The authors focus on addressing the issue of liberal inference (elevated Type I errors) shown by typical inference techniques under resampling. They note that a number of inference procedures, such as the t-test, incorporate the degrees of freedom as a parameter. The degrees of freedom refers to the number of independent pieces of information that can be used to estimate a parameter of interest (e.g. the EPE or $Var[EPE]$). Normally the number of independent pieces of information would equal the number of PE values. However, as previously discussed in the case of data reuse the PE values are not independent. Therefore the number of PE values does not indicate the number of independent pieces of information. The t-test uses the degrees of freedom to account for the fact that the $Var[EPE]$ is calculated from the sample variance of the PE values rather than the population variance. Compared to the z-test which assumes the EPE is normal distributed the t-test assumes the EPE follows the Student's t distribution parametrized by the degrees of freedom. The Student's t distribution is similar to the normal distribution, approaching it as the degrees of freedom increase. When the degrees of freedom are small the distribution has fatter tails, leading to more conservative inference, reflecting the fact that the standard deviation may be incorrect. Therefore the authors argue that the problem of liberal inference can be fixed by still calculating the sample standard deviation as normal but by providing the correct degrees of freedom for use within a t-test (or its paired version).

Unfortunately there is no known method for correctly calculating the number of independent pieces of information within the J correlated PE estimates. If such a method was available then it stands to reason that the correlation would also be known and the variance corrected accordingly through formula discussed in [141]. Therefore [26] empirically evaluate the effects of lowering this parameter through experiments where the outcome of statistical tests (liberal or conservative) could be measured. Varying the degrees of freedom from the number assuming independent PE's the authors show that

[8]Recall that in this work the prediction error function has been motivated as the binary decision function over the truncated average discrete Fréchet distance.

correct inference is achieved by using a much lower number of degrees of freedom, as was expected from the theoretical motivation. Based on their empirical results they conclude with the recommendation that the likelihood of incorrect inference can be reduced by assuming the PE's only represent 10 pieces of independent information and therefore fixing the degrees of freedom to 10 under 10 fold cross validation with 10 permutations. While this approach may seem tempting due to its simplicity, it should not be used since as previously discussed altering the degrees of freedom simply alters the Students t-distribution being used to test against, with a similar effect of attributing a larger variance to the sample. However, as noted [16], there is no general way to account for the dependence and so the use of a single parameterization to *fix* this dependence is not possible, meaning that the empirically arrived at value of 10 degrees of freedom is likely only valid for that and potentially similar data sets. In this light of particular note is the investigation in [26] into the degrees of freedom under the corrected resampling procedure from [141]. Using the corrected resampling procedure it is noted that the degrees of freedom that obtain Type I errors at the level of significance increases consistently with the increase in the number of runs and that this level is only approximately 10% lower than the expected, indicating that the variance correction mostly accounted for the elevated Type I errors. This note attests to the fact that *both* adjustments are not required in general.

Overestimating the variance

An alternative to attempting to estimate the variance with a small bias is to argue that conservative inference is much better than liberal inference and develop an overestimate of the variance. This argument is easily made by noting that research aims to develop predictors of superior performance. Therefore if conservative inference is used claims of predictor superiority (for a given level of statistical significance) is preserved at the expense of algorithms that may provide some performance increase but not a very large amount. If liberal inference is used, however, this claim is not preserved and many predictors of minimal gain will be reported. This is not desired.

The approach to the overestimation of the variance comes from [141], the same paper as the alternate method of variance estimation with a goal of minimizing the bias (see section 3.2.3, equation 3.15). The authors make the following conjecture:

Conjecture 1. *In most situations* $Var[EPE^{|\mathcal{T}|}_{|\mathcal{K}|}]$ *should decrease in* $|\mathcal{K}|$. *[141, pg 245]*

and based on this conjecture, the assumption:

Assumption 1. *$Var[EPE^{|\mathcal{T}|}_{|\mathcal{K}|}]$ is decreasing in $|\mathcal{K}|$*

This assumption is then used to form their over-estimation of the variance.

Recall the subscript on the EPE denotes the size of the training set and the superscript the size of the test set. The argument behind the conjecture is based on the observation that the variability in the $EPE^{|\mathcal{T}|}_{|\mathcal{K}|}$, given a fixed number of samples (J) and $|\mathcal{T}|$ is dependent only on the training set. Considering this, it is required to show that as $|\mathcal{K}|$ decreases the $Var[EPE^{|\mathcal{T}|}_{|\mathcal{K}|}]$ increases. While not proven (hence it is only a conjecture) it is argued that intuitively more training data will proved the algorithm with information that enables more consistent predictions across the space of possible inputs hence leading to lower variability in the overall mean error scores. In the case of route prediction algorithms this seems reasonable since extra training data is expected to lead to an increase in performance, reducing the number of wildly incorrect predictions, hence reducing the variance.

Assuming that $Var[EPE_{|\mathcal{K}|}]$ does decrease in $|\mathcal{K}|$ they propose the following approach to overestimate the variance of interest. The approach is based on splitting the total data set (D) in half and using each independently to arrive at two independent estimates of the EPE based on this smaller set size. In the work of [141] the EPE is arrived at using a form of cross validation that randomly samples without replacement to select a training set of size $|\mathcal{K}|_D$ using the remaining observations as the test set ($|\mathcal{T}|_D = |D| - |\mathcal{K}|_D$). The subscripts denotes the data set for which the training/test set size belongs to. This is then repeated a fixed number of times, J, generating J PEs. The EPE is then the average of the PEs.

Let the two new data sets be denoted D_{1a} and D_{1b} and each contain a randomly selected set of observations (without replacement) from D such that $|D_{1a}| = |D_{1b}| = \lfloor \frac{|D|}{2} \rfloor$ and let ${}_{1a}EPE^{|\mathcal{T}|_D}_{\lfloor \frac{|D|}{2} \rfloor - |\mathcal{T}|_D}$ and ${}_{1b}EPE^{|\mathcal{T}|_D}_{\lfloor \frac{|D|}{2} \rfloor - |\mathcal{T}|_D}$ denote the EPE for each data set of size $\lfloor \frac{|D|}{2} \rfloor$ respectively. Recall once more that the subscript on the EPE denotes the size of the training set $|\mathcal{K}|$ and superscript, $|\mathcal{T}|$, denotes the size of the test set. Note that despite splitting the data set in half, the size of the test set remains the same, only the training set size has been reduced.

These two EPE, ${}_{1a}EPE$ and ${}_{1b}EPE$ are then both *independent* and provide a unbiased, but noisy, estimate of the variance of the $EPE^{|\mathcal{T}|_D}_{\lfloor \frac{|D|}{2} \rfloor - |\mathcal{T}|_D}$ via the standard formulation

for variance:

$$Var\left[EPE^{|\mathcal{T}|_D}_{\lfloor\frac{|D|}{2}\rfloor-|\mathcal{T}|_D}\right] = \frac{\left({}_{1a}EPE^{|\mathcal{T}|_D}_{\lfloor\frac{|D|}{2}\rfloor-|\mathcal{T}|_D} - {}_{1b}EPE^{|\mathcal{T}|_D}_{\lfloor\frac{|D|}{2}\rfloor-|\mathcal{T}|_D}\right)^2}{2} \tag{3.17}$$

However, as noted this is noisy estimate. To address this the process of generating ${}_{1a}EPE^{|\mathcal{T}|_D}_{\lfloor\frac{|D|}{2}\rfloor-|\mathcal{T}|_D}$ and ${}_{1b}EPE^{|\mathcal{T}|_D}_{\lfloor\frac{|D|}{2}\rfloor-|\mathcal{T}|_D}$ is repeated. Due to the random selection of observations to make these sets, two different data sets are obtained and the process repeated M times. This leads to M unbiased estimates of $Var[EPE^{|\mathcal{T}|_D}_{\lfloor\frac{|D|}{2}\rfloor-|\mathcal{T}|_D}]$ which can be averaged to increase the accuracy of the estimate. In summary this leads to the following estimate of the variance:

$$Var\left[EPE^{|\mathcal{T}|_D}_{\lfloor\frac{|D|}{2}\rfloor-|\mathcal{T}|_D}\right] = \frac{1}{2M}\sum_{m=1}^{M}\left({}_{ma}EPE^{|\mathcal{T}|_D}_{\lfloor\frac{|D|}{2}\rfloor-|\mathcal{T}|_D} - {}_{mb}EPE^{|\mathcal{T}|_D}_{\lfloor\frac{|D|}{2}\rfloor-|\mathcal{T}|_D}\right)^2 \tag{3.18}$$

which is not the estimate of the variance of interest, $Var\left[EPE^{|\mathcal{T}|_D}_{|\mathcal{K}|_D}\right]$, but making the assumption that follows from conjecture 1, is an overestimate of $Var\left[EPE^{|\mathcal{T}|_D}_{|\mathcal{K}|_D}\right]$ as desired. In their work [141] evaluate the performance of the variance estimate through a simulation study considering a range of algorithms including linear regression, decision trees and first nearest neighbour over problems of classification and regression. Evaluating the estimate as part of the Z-test the results indicated that the estimator does produce conservative inference.

Based on the theoretical work of [141] it is possible to consider another overestimate of the variance. Of note is that in [141, page 245, proposition 2] it is shown that $Var[EPE^{|\mathcal{T}|}_{|\mathcal{K}|}]$ is non-increasing in $|T|$. Therefore it is possible to make the over-estimation of the variance benefit from the additional assumption that $Var[EPE^{|\mathcal{T}|}_{|\mathcal{K}|}]$ is increasing in most situations as $|T|$ decreases. This is achieved by also reducing the size of test set as well as the training set. Note that due to proposition 2 from [141] the reduction in the test set size can only increase the variance or have no effect. Therefore conjecture 1 is relaxed somewhat to:

Conjecture 2. *In most situations the change in* $Var\left[EPE^{|\mathcal{T}|}_{|\mathcal{K}|}\right]$ *should show a net increase from the simultaneous reduction of both* $|\mathcal{K}|$ *and* $|\mathcal{T}|$.

leading to the over-estimator of this variance estimation being dependent on the assumption:

Assumption 2. *The simultaneous reduction of both* $|\mathcal{K}|$ *and* $|\mathcal{T}|$ *to results in a net non-decreasing change in* $Var\left[EPE^{|\mathcal{T}|}_{|\mathcal{K}|}\right]$.

The corresponding estimator based on 10-fold cross validation (where each training set comprises of 90% fo the data with 10% used for testing) is then:

$$Var[EPE_{\frac{9}{10}\lfloor\frac{|D|}{2}\rfloor}] = \frac{1}{2M}\sum_{m=1}^{M}\left({}_{ma}EPE^{\frac{1}{10}\lfloor\frac{|D|}{2}\rfloor}_{\frac{9}{10}\lfloor\frac{|D|}{2}\rfloor} - {}_{mb}EPE^{\frac{1}{10}\lfloor\frac{|D|}{2}\rfloor}_{\frac{9}{10}\lfloor\frac{|D|}{2}\rfloor}\right)^2 \tag{3.19}$$

From the equation it is clear that in order to use all data as the test set is reduced the training set is expanded compared to the estimator in equation 3.18. Importantly the training set size here is still significantly smaller (50% to be exact) than the training set of the variance for which the over-estimate is desired. In contrast to the estimator from equation 3.18 this estimator will still provide the desired over-estimate when the assumption 1 is violated, providing assumption 2 still holds true. Since neither assumptions have been proven the choice between estimators is somewhat arbitrary. In this thesis the second is preferred due to its inclusion of the reduced test set size for which the $Var[EPE]$ is known to be at least non-increasing in $|\mathcal{T}|$. A more through investigation into the assumptions and their correspondence is interesting future work.

Note that a third variant of the over-estimator where only the test set size is decreased is not possible since after halving the data each subset does not have enough observations to maintain a fixed training set size (typically $70-90\%$ of the original data size).

In all cases a choice must be made for M. Examining simulations [141] note that small (< 5) M can lead a poor approximation resulting in liberal inference. Higher values of M lead to more conservative inference. They conclude by recommending $5 \leq M \leq 10$. Throughout this thesis M is set to 7.

The analysis of correlated data: Alternate approaches

Other approaches to the analysis of correlated data exist in the literature, most notably in the field of geospatial science and ecology where generalized estimating equations or spatial autocorrelation in generalized linear mixed models are used. Unfortunately the approaches can not be directly applied in the case of evaluating resampled data. With regard to generalized estimating equations (GEEs) the robust estimator requires the definition of clusters which are independent, while allowing for only approximate specification of the correlation within clusters (see [86] for an accessible introduction to GEEs). Generalized mixed models on the other hand required a distance function to be specified between observations, which is not easily provided since the distance is used to

take into account the correlation which is dependent not only on the test object/set but also on the training set for each test unit in the pair of test units being considered.

Summary of approaches to calculate the variance

The discussion on calculating the variance has noted that in general the variance is under-estimated leading to liberal inference which is undesired, with conservative inference preferred. This ensures that any prediction algorithm reported as being superior actually is. Addressing this a number of approaches to correcting the variance estimate have been put forward. These can be generally grouped into two categories, ones that attempt to provide an estimator as close as possible to the real variance accepting that liberal inference will sometimes occur and those that aim to over-estimate the variance ensuring no liberal inference occurs. Since conservative inference is preferred in this context the latter is recommended. Two over-estimators of the variance were discussed. Of these it was noted that both are unable to guarantee over-estimation. However, the approaches are based on assumptions highly likely to be true and results from [141] provide strong empirical evidence that this is the case. Making only slightly different assumptions the choice between the two is debatable and either could be used. In this thesis the second estimator is used. The reason is two-fold. Firstly, as previously mentioned, the variance is still able to over-estimate in certain situations where one assumption fails. The second is purely practical, with the estimator being slightly easier to implement.

3.2.4 Empirical observations: Variance estimators in evaluating movement predictors

In the previous section three methods for estimating the variance were primarily discussed. While a full investigation into their use in the context of evaluating route prediction is beyond the scope of this thesis and left for future work, some preliminary investigations were undertaken to provide some empirical observations.

Of the three methods one attempted to provide an estimate that was as close as possible to the real variance and two that aimed to over-estimate the variance to ensure conservative inference. The first method is abbreviated to NB1 (detailed in equation 3.15). Of the two that aimed to overestimate the variance, the one developed by [141] which only reduces the size of the training set is denoted NB2 (detailed in equation 3.18). The

third method, a variant of the second developed in this thesis, is denoted OET (detailed in equation 3.19). It is of note that in both NB1 and NB2 random resampling without replacement was used to select the training and test sets as per the paper [141]. In the OET estimator repeated 10-fold cross-validation is used. This was chosen due to its general recommendation as a estimator of the EPE in section 3.2.2. Note that it is not possible to use repeated 10-fold cross validation in the NB2 estimator since only the training set is reduced in size. In all cases 100 training and test sets were used.

The investigation primarily had two aims. The first was to consider the difference in the variance as calculated by the three different methods. This was considered important since the authors proposing estimators NB1 and NB2 note that when the correlation between the predictors being compared is high (> 0.1) then the estimator E1 tends to over-estimate the variance and vice versa. Of course this is dependent on the predictors and the data and hence the requirement to consider the problem in context. The second aim was to provide a some preliminary observations in the difference between the similar NB2 and OET estimators.

In order to provide preliminary observations into the variances calculated by the three methods two predictors from chapter 6 were selected. Two predictors are required as in practice the prediction error *difference* between two predictors is used as the error measure rather than the prediction error itself. This is further discussed later in section 3.2. The result is an estimate of the variance for each of the three methods. In the case of NB2 and OET there is the additional parameter M, which is varied between 5 and 10. Note that the effect of varying M is not an aim of the investigation due to the small sample considered and the examination provided in [141]. The results are shown in table 3.1.

The results of the preliminary study shows that the NB1 estimator in this case always provides a higher estimate of the variance than NB2 and OET. Since NB1 was meant to provide a estimate as close as possible to the true value it may seem counter-intuitive that the estimator provides a larger variance than the estimators aiming to overestimate the variance. However, this is simply explained by considering that the two predictors examined here are somewhat similar with the two methods only varying in the information they used to make the prediction (see section 6.5, the two predictors were *CT:XY* and *CT:XY-SEQ*). Considering this in conjunction with the knowledge that, by construction, the estimator will overestimate the variance if the predictors are highly

Estimator	Sampling Type	parameters	variance
NB1	Resampling ×100, 10% test set, 90% training set		0.0002699165
NB2	Resampling ×100, 10% test set, 90% training set	$M = 5$	0.0001626528
NB2	Resampling ×100, 10% test set, 90% training set	$M = 7$	0.0001831969
NB2	Resampling ×100, 10% test set, 90% training set	$M = 6$	0.0001831969
NB2	Resampling ×100, 10% test set, 90% training set	$M = 8$	0.0001767273
NB2	Resampling ×100, 10% test set, 90% training set	$M = 9$	0.0001727637
NB2	Resampling ×100, 10% test set, 90% training set	$M = 10$	0.0001695928
OET	10× 10-fold cross-validation	$M = 5$	0.0001123376
OET	10× 10-fold cross-validation	$M = 6$	0.0002141965
OET	10× 10-fold cross-validation	$M = 7$	0.0002132998
OET	10× 10-fold cross-validation	$M = 8$	0.0001912894
OET	10× 10-fold cross-validation	$M = 9$	0.0001877227
OET	10× 10-fold cross-validation	$M = 10$	0.0001695732

Table 3.1: Table showing estimates of the Var[EPE] between two predictors from chapter 6 under the different variance estimators discussed in section 3.2.3.

correlated the result is actually not that unexpected. Since comparisons of this type form a core part of investigations in this thesis, the observation lends additional weight to the argument that the NB1 estimator may not as suitable as the other estimators.

Considering the differences between the NB2 and OET estimators, in four out of six cases, the OET estimator provides larger estimates of the variance. The opposite occurs when $M = 5$ and $M = 10$, with only a very small difference ($1.96e - 08$) when $M = 10$. Given the preliminary nature of this investigation no firm conclusion can be drawn, however, it seems that if anything the difference made by choosing M has a greater effect than the choice between NB2 and OET. This observation is not unexpected due to the two estimators being close variants of one another. Therefore the preliminary investigation does not alter the previous recommendation, the use of the OET, for estimating the variance.

3.2.5 Tests for comparing predictors

Tests for comparing two predictors

Since the resampling is controlled by the experimenter it is possible to obtain paired prediction error scores over all algorithms. This is desirable since paired tests are more powerful and correctly take into account that the same training and test sets are used in all predictors. Failure to take this into account also typically leads to inflated Type I error (over and above previously discussed Type I error inflation) [206]. Therefore paired tests are recommended. Using a correct estimate of the variance as motivated in section 3.2.3 along with the mean motivated in section 3.2.2 enables the direct application of either the paired t-test or the paired difference test. Recall that the previously discussed mean was the EPE, defined in equation 3.14 as:

Recall $\mathcal{K}$ denotes a training set of historic trail observations

Recall $\mathcal{T}$ denotes a test set of $\langle E_i, F_i \rangle$ pairs

Let $\mathcal{Q} = \{Q_1, \ldots, Q_{|\mathcal{Q}|}\}$ be a set pairs of test and training set pairs, $Q_j = \langle \mathcal{K}_j, \mathcal{T}_j \rangle$

Recall $\mathcal{P}$ denotes a predictor

Recall $PE(\cdot, \cdot)$ is the prediction error for a trained predictor over the given test set

defined in equation 3.11

Then:

$$EPE_{|\mathcal{K}|}(\mathcal{P}, \mathcal{Q}) = \frac{\sum_{i=1}^{|\mathcal{Q}|} PE(\mathcal{P}(\mathcal{K}_i, \cdot), \mathcal{T}_i)}{|\mathcal{Q}|}$$

The paired test simply replaces the PE scores with the paired difference between PE scores from two predictors, with the pairing based on the training and test set used. Specifically, consider two predictors, A and B. The paired mean (denoted here as the $DEPE$) is then defined as:

$$DEPE_{|\mathcal{K}|}(A, B, \mathcal{Q}) = \frac{\sum_{i=1}^{|\mathcal{Q}|} PE_A(\mathcal{P}(\mathcal{K}_i, \cdot), \mathcal{T}_i) - PE_B(\mathcal{P}(\mathcal{K}_i, \cdot), \mathcal{T}_i)}{|\mathcal{Q}|} \tag{3.20}$$

Where PE_X denotes the PE score from predictor X for the given training and test set.

The corrected variance is calculated as per equations 3.15, 3.18 and 3.19 for the estimators $NB1$, $NB2$ and OET respectively by replacing the EPE with the $DEPE$ and any

PE scores with the paired difference scores as required.

In contrast to the paired difference test which is based solely on the mean and sample variance, the paired t-test adjusts the test making it more conservative via a parametrization in the number of degrees of freedom. Unfortunately, as discussed in section 3.2.3, the degrees of freedom is not known. When the $NB2$ estimator of the variance is used, which produces an over-estimate of the variance given the previously discussed assumptions, [141] argue that the paired t-test is not required since the estimate is already conservative. However, the paired t-test is still valid, assuming conservative inference is acceptable as it simply produces more conservative inference. This can be seen by considering that the Student's t distribution approaches the normal distribution as the degrees of freedom becomes large and noting that the Student's t distribution has larger area in the tails for lower degrees of freedom compared to the normal distribution. Therefore the paired difference test is recommended, however, if desired the paired t-test can be used to make the inference more conservative. In any case the fact that the inference is conservative must be highlighted and the implications noted due to the use of the over-estimated variance.

Tests for comparing multiple predictors

It is generally desirable to compare more than two predictors at one time, which is known as multiple hypothesis testing. Multiple hypothesis testing is a well known statistical problem. The issue is that as the number of comparisons increases, so does the likelihood of observing a difference purely due to chance. Known as the *family-wise error* it is important that this is controlled. The most common way is via adjusted p-values. In this case the p-values are computed for all pairwise combinations of the predictors of interest and then an adjustment procedure performed in order to control the family-wise error. The result is a set of p-values that can be interpreted as normal.

To calculate adjusted p-values numerous techniques exist. [70] provide a good description of such procedures in the context of comparing classifiers over multiple datasets, recommending the Bergmann-Hommel procedure. Noting its computational complexities Shaffer's procedure is recommended in the case the computation can not be done, noting that the latter suffers from lower power. These recommendations additionally apply in the case of analysis from dependent samples provided the p-values have been generated via a corrective method. Therefore this recommendation is repeated here in

the case of analysis of movement predictors.

3.2.6 Summary: Recommended tests for the hypothesis testing between two predictors

Hypothesis testing is a broad process with many different approaches. Broadly speaking the goal was to compare the mean performance of two predictors. The approach recommended was a paired difference test using a over-estimate of the variance. The the null hypothesis was that there was no (zero) difference between the mean performance of the two predictors. Mean performance was defined as the mean prediction error independent of the training and test sets used. Since limited data generally exists to perform these tests data re-use was recommended to provide a better estimate of the mean performance of each predictor. Specifically repeated ten fold cross-validation was recommended with 10 repeats. Unfortunately when re-using data the standard calculations of the variance provides under-estimates which can lead to liberal inference (indicating significant differences exist when they do not in reality). Therefore alternative methods for calculating the variance was discussed, concluding with the recommendation of methods that over-estimate the variance preferring conservative inference over liberal inference.

Generalizing to comparisons involving more than two predictors the Bergmann-Hommel procedure was recommended to adjust the p-values calculated from the pairwise comparisons of all predictors of interest. In all cases normality should be checked for, but unless small sample sizes are used, this will almost invariably hold.

3.3 Descriptive statistics and visualizations

In section 3.1 the use of a binary error function was proposed for measuring movement prediction performance. In section 3.2 statistical tests were detailed to allow the comparison of multiple predictors based on their generalization error for a given value of the relevance parameter. Reporting only that a statistically significant difference was observed or not, however, is not sufficient and does not convey all the information available or desired. For instance, the question of which value should be selected as the relevance parameter within in the error function is somewhat subjective and may have an important impact on the results. In general it is important to additionally provide a

visualization of the data to portray the relative difference between algorithms. To this end three graphs to complement the proposed error function and statistical tests are proposed for use in evaluating movement prediction algorithms.

The first proposed graph is specific to the error function based on the relevance parameter, plotting the EPE vs the parameter for a predefined parameter range. Depending on the application there is often a maximum range at which performance would be unacceptable, but performance within tighter bounds is often an important consideration. Additionally the graph provides a good overview of competing predictors' performance when graphed together. An example is shown in figure 3.11. From a practical perspective these graphs require the error from individual test is recorded during the evaluation phase and the conversion to binary relevance done on-the-fly in statistical testing and in the generation of descriptive statistics and graphs. In the **R** statistical project for computing software [158] this graph can be generated with the in-built *plot* command.

The second graph is required to aid in the interpretation of the statistical tests and is common to analysis in many domains. The statistical inference provides an indication as to the presence of a real difference (or not) between the EPE values of two different predictors, but not which is better, or by how much. To ascertain this information and visualize the information the tests used to arrive at this result a side-by-side box plot of the PE values is recommended. An example is shown in figure 3.12 where additionally notches have also been added to the side of the boxes in order to provide an at-a-glance indication of significant results - if the notches of two plots do not overlap this is strong evidence that the two medians differ to a statistically significant level [34]. This is to be taken as a very rough guide as the calculations do not take into account the dependency issues discussed in section 3.2. Accompanying the box plot should be a table with at least the mean (and potentially standard deviation) or this data written on the graph, since it is not portrayed otherwise. In the **R** statistical project for computing software [158] this graph can be generated using the in-built *boxplot* command and the means added via the *mtext* command.

A third graph is proposed to visually display the truncated average discrete Fréchet distance (see section 3.1) of the individual observations to provide a overview of the individual predictions. This is important at least in the exploratory stage and can provide important comments on the data. For this side-by-side histograms of the counts of the truncated average discrete Fréchet distance per predictor is advised. Since the

graphs tend to have long tails (see figure 3.13), subsections of the graph can be zoomed in on to provide further comment. An example is shown in figure 3.14. In **R** [158] this is easily achieved using the *lattice* package [169], where side-by-side plots are created with the single command *histogram* and data subsets selected via the *subset* command.

3.4 Evaluating movement prediction from GPS logs: A summary

In this chapter different evaluation metrics, testing methodologies and statistical tests were examined due to a lack of coherence within the literature with the aim of providing a practical guide to evaluation in the domain of movement prediction. A summary the recommendations is presented below. Included in the summary are notes indicating the availability of the methods as ready to use function within the **R** statistical project for computing software [158].

- **Test statistic for a single prediction:** The truncated average discrete Fréchet distance followed by a parametrised binary function of relevance. This is required due to the vast discrepancy between the range of values useful predictions fall into versus useless predictions, with useless predictions dominating the measure otherwise. Additionally the binary function is simply parametrised and easily interpretable. See section 3.1.9. Code to calculate the truncated average discrete Fréchet distance in the **R** statistical project for computing software [158] is provided in appendix A.

- **Testing methodology:** Repeated k-fold cross-validation. This is required in order to fully use the limited data available and calculate the generalized prediction error of the predictor for arbitrary training sets for the data. Repeated k-fold cross-validation is easily implemented, see section 3.2.

- **Hypothesis testing:** Pairwise comparisons of predictors via paired difference tests using the OET variance estimator (defined in equation 3.19) and adjusted in the case of more than two predictor via the Bergmann-Hommel procedure. The OET estimator of variance rather than standard calculations is required since the re-use of the data in multiple training/test set pairs means that standard variance calculations are not accurate and lead to liberal inference where significant

differences that do not exist at the specified significance level are reported. The formula for the OET is provided in equation 3.19 and is easily implemented. Unfortunately the Bergmann-Hommel procedure is currently not implemented in the **R** Statistical project for computing software. Of those available in **R** the Holm procedure from the **R** package *muToss* [140] is recommended. Compared to the Bergmann-Hommel procedure the Holm procedure is more conservative [70] once more ensuring that results reported to be significant are significant at or below the specified significance level.

- **Descriptive statistics:** Three graphs were proposed, one examining the choice of the error function parameter, one supporting the hypothesis testing and one providing an overview of the individual prediction level scores under the average Fréchet distance. All graphs are easily constructed using **R** Statistical project for computing software. See section 3.3.

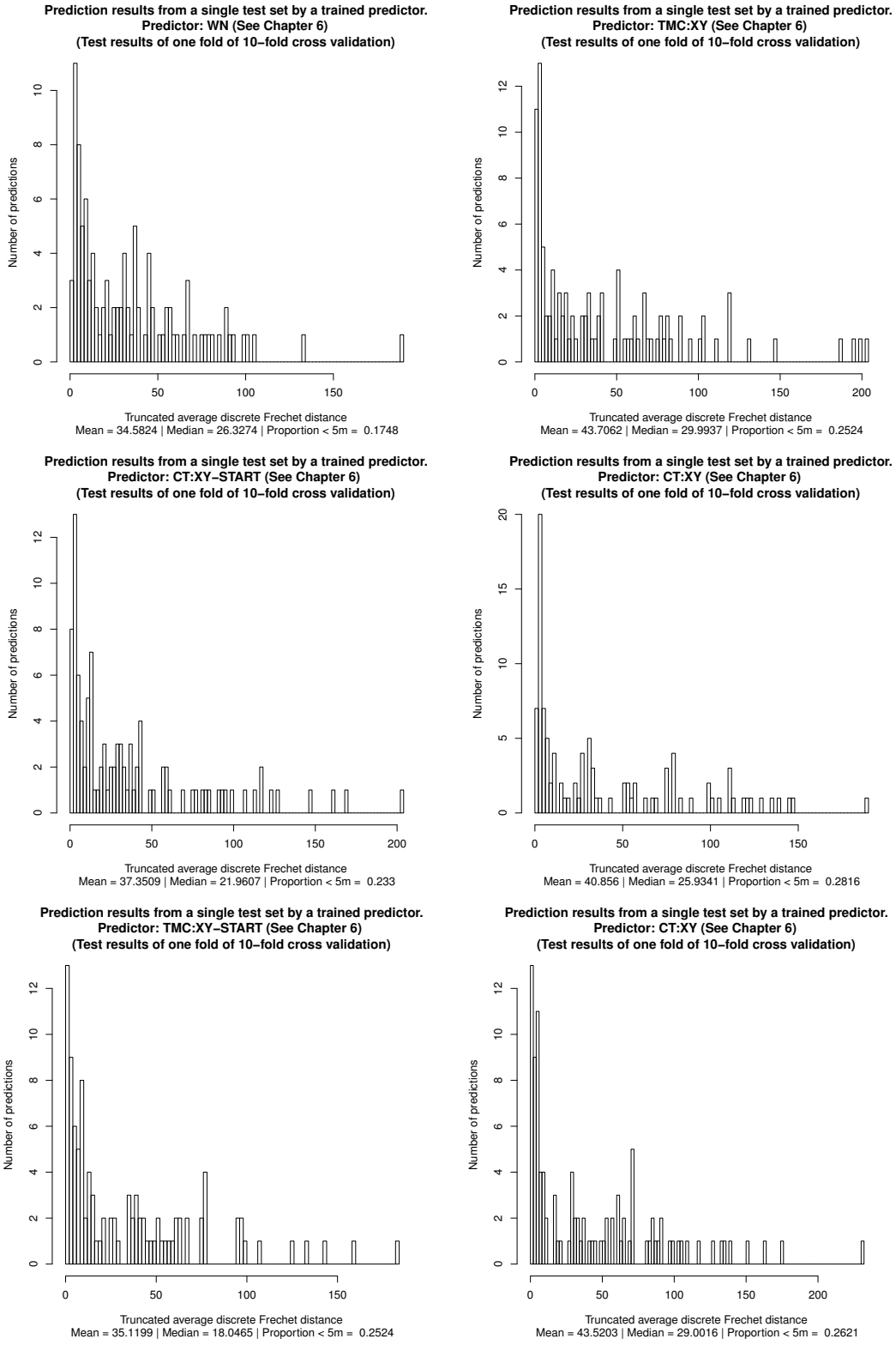


Figure 3.10: Graph showing histograms of individual predictions (truncated average discrete Fréchet distance) for a six trained predictors for a single test set. Note that the histograms are *not* normally distributed, with all being positively skewed. The figure also highlights real world examples where the median for predictor A is smaller than the median of predictor B while predictor B has predicted more below a relevance threshold of $5m$ than predictor A. The first column represents predictor A and the second predictor B. Each row then represents a comparison of interest.

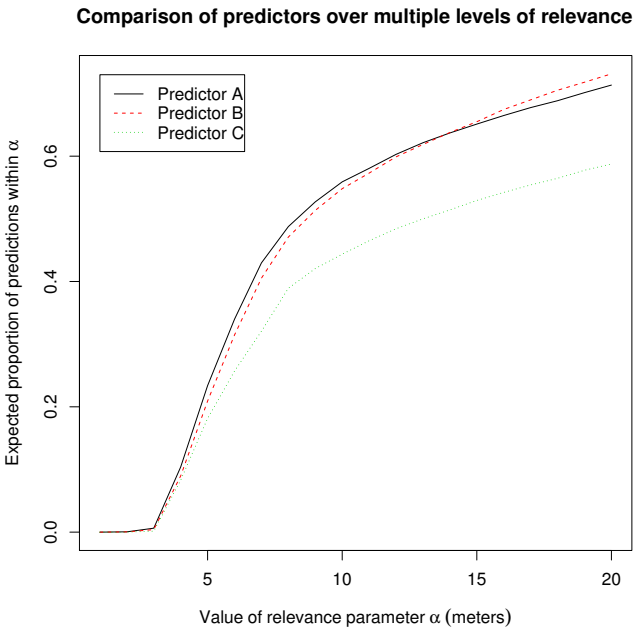


Figure 3.11: Graph showing the performance of three predictors (A, B and C) with varying settings of the relevance parameter in the error function.

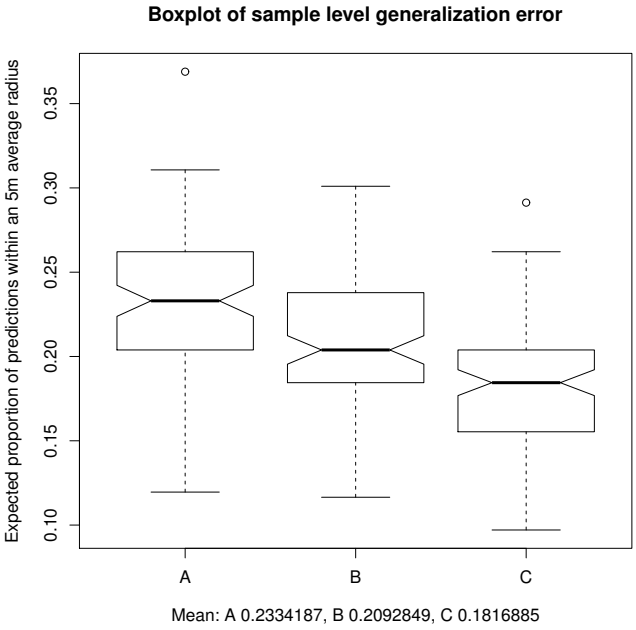


Figure 3.12: Box plot showing the sample level generalization error

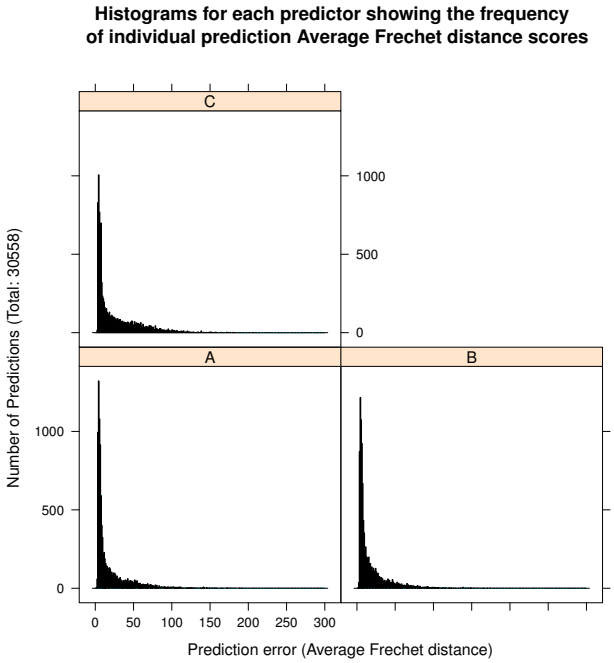


Figure 3.13: An example side-by-side histogram plot showing three predictors (A, B, C)

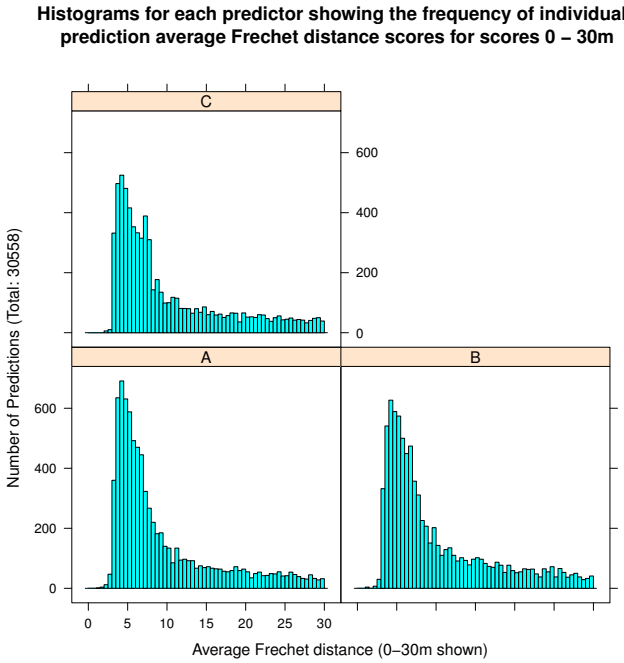


Figure 3.14: An example side-by-side histogram plot showing a subset of the data (for readability) for the three predictors A, B and C

Chapter 4

Efficient Prediction: a Naive Bayes' Model

In achieving the vision of intelligent mobile agents it is desirable to run prediction algorithms locally on mobile devices which are resource limited. To this end this chapter contributes a computationally inexpensive algorithm for predicting pedestrian movement which, under certain realistic conditions is empirically shown to show no significant difference in prediction accuracy in comparison to a more complex and significantly more computationally expensive current state-of-the-art approach.

In contrast to current computationally efficient predictors such as the basic fixed length Markov model introduced in chapter 2 (section 2.1.1), which models sequence explicitly, this chapter proposes a model based on a relaxed notion of sequence providing a theoretical motivation as to how such an approach can allow for additional modelling mechanisms which can provide a large improvement in prediction accuracy. This is then confirmed empirically in section 4.4. It is important to note that the proposed approach is still Markovian in nature in that it only maintains probabilities conditioned on single states, hence retaining the properties of cheap runtime and space complexity. Such approaches are in contrast to higher order models (Bayesian or otherwise) which consider future states conditionally dependent on as many past states as either modelled (Dynamic Bayesian networks) or known (variable length Markov Models, Universal predictors and most pattern matching approaches). These higher order models potentially provide better prediction accuracy but at much higher computational costs (see chapter 2, see sections 2.1.2 and 2.1.3) which can limit or prevent their ability to perform in real time in resource limited mobile devices. In contrast to Bayesian networks or

pattern matching approaches this chapter proposes a novel algorithm that is based on Bayes' rule. Approximating the probability of future states conditioned on all known past states under an assumption of independence, the approach is motivated by the surprisingly good real world classification results of naive Bayes' classifiers [24, 151] in domains such as image classification. The chapter concludes with results showing by additionally utilizing specific encodings (for example, encoding direction) no significant difference in prediction accuracy in comparison to the more complex models in real world conditions.

4.1 Motivation

Movement prediction has a wide range of potential applications. Traditional applications have involved movement prediction for efficient mobile phone service provisioning (e.g. [38, 148]) among others for which a decent amount of computation power can be assumed[1]. More recently, however, applications relating to *intelligent agents* have emerged, primarily embedded in mobile consumer devices with built-in navigational technologies and consequently with limited resources to varying degrees. Recent applications have included the development of an intelligent navigation aid to alert people with cognitive disabilities when they had deviated from their normal route [150], in car systems aiming to provide ahead-of-time notifications about upcoming traffic hazards or points of interest [69], systems to help improve hybrid vehicle efficiency [102] and services to allow ad-hoc networking [33]. In light of these applications, in which there are many cases where the application is desired to run locally and on limited hardware, an algorithm with low runtime complexity and fixed known memory usage is proposed. The novel approach fills the gap between the simplistic, but generally underperforming, first order Markov Model and the majority of prediction approaches which provide superior prediction accuracy. The former has a low fixed memory requirement in the order of the number of discrete locations and low runtime costs while the latter has memory and/or computational requirements in the order of the number of spatially different observations in the entire, or a selected subset, history. This is achieved by *approximating* the full order Markov chain and through the use of a specific encoding that addresses a number of drawbacks such an approximation presents. The approach maintains Markovian properties and hence the same level of low resource usage as traditional Markovain

[1] Although not necessarily required, see [182] and the discussion in chapter 2.

approaches but demonstrates significantly superior prediction accuracy, nearing the full order models under certain realistic conditions.

4.2 Model definition

Conceptually the proposed model relaxes the notion of sequence from previous Markovian approaches. It is of note that previous models always attempt to model the probability of a spatial region r_1 of interest given another spatial region r_2 as the number of times the first appears *directly* after the second in some ordered time sequence. This can be clearly seen in the formal definition of Markov models (see chapter 2, section 2.1.1) and this extends to all Universal Predictors and a number of pattern matching algorithms (see sections 2.1.2 and 2.1.3). In contrast the proposed model relaxes the notion of sequence encoding in favour of a more holistic approach. Specifically the proposed model builds conditional probabilities based on the number of times a region of interest *occurs in the future* with respect to a conditioning spatial region. Therefore the approach relies on the holistic view of the data given by the particular partitioning of the raw GPS data into trails, rather than the exact sequencing information.

Before continuing it is worth mentioning that the formulation and hence symbols used here are different to those from the general problem definition of route prediction presented in section 1.1. Specifically here the trails are defined by spatial regions rather than point instances. The formulation still addresses the problem of interest, with the conversion of the point instances to spatial regions able be achieved in the basic case via grid based quantization of the spatial area of interest.

Let a trail be defined as a temporally-indexed set of spatial regions ($H = \{\mathsf{h}_1, \ldots, \mathsf{h}_{|H|}\}$) over $X \times Y$ metre quantized area of spatial interest ($S = \langle s_1, \ldots, s_{X \times Y} \rangle$) and the notation of D to represent all historic trails, $H_1, \ldots, H_{|D|}$, formally the approach retains a matrix of the form:

$$\forall s_i, s_v \in S, P(s_v|s_i) = \frac{\sum_{j=1}^{|H|} \begin{cases} 1 & \text{if } \exists(\mathsf{h}_w = s_v \wedge \mathsf{h}_q = s_i | \mathsf{h}_w \in H_j \wedge \mathsf{h}_q \in H_j \wedge q < w) \\ 0 & \text{otherwise} \end{cases}}{\sum_{j=1}^{|H|} \begin{cases} 1 & \text{if } \exists(s_i \in H_j) \\ 0 & \text{otherwise} \end{cases}} \tag{4.1}$$

Informally, and with slight abuse of notation, it can be seen that the aim is to model the probability that s_i occurs in the future, given s_v occurs in the past. I.e.

$$\forall s_i, s_v \in S, P(s_{i \text{ in future}} | s_{v \text{ in past}}) \tag{4.2}$$

Note that such an encoding leads to a different question during inference. In traditional Markovian modelling the question being asked is *what is the probability of being in a spatial region next, given the current region?* whereas the new encoding asks the question *what is the probability of being in a spatial region anytime in the future, given the current region?*.

It is of note that next step and route prediction can be performed in either case. In the former, prediction must occur via recursion, i.e. after making the prediction, the prediction becomes the history from which the probabilities for the next step are predicted from. In the latter, this does not have to be the case, although it can be. This is further discussed in section 4.3, although in summary this is not done since this merges knowledge which is known to be true (the route taken so far) with guesses as to future locations (the next step prediction for instance) to then make further predictions. Since the incorporation and then re-prediction based on single poorly predicted location can compound the prediction errors an alternative route prediction mechanism is implemented that reasons solely on the known route taken so far.

Before continuing the instinctive intuition that the information lost in the relaxed encoding can lead to errors is discussed and confirmed, additionally noting how this can be reduced in practice. Later in section 4.4 empirical evidence is given showing that the gains in prediction accuracy can significantly outweigh the losses incurred by these errors.

The relaxation of the model to record the probabilities conditioned on the presence of a variable in a variable length history rather than the previous timestep introduces the possibility that loop-like trails can artificially boost the probability. This is shown in figure 4.1. The circles represent one trail which has been seen one hundred times in the historic set of paths and the triangles another trail that has only been seen once. The current state is the grid square containing the diamond. In traditional Markovian approaches the clear choice would be the first in the sequence represented by the circles. This intuitively seems to be the correct answer, whereas the proposed approach would select the grid square occupied by the second green triangle as the next movement.

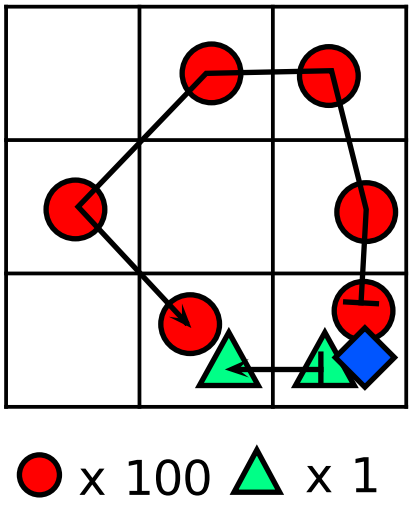


Figure 4.1: An example showing a case where a traditional Markovian approach will outperform the proposed approach.

Having considered an example where the traditional approach should outperform the proposed approach, consider the following example where the reverse occurs, shown in figure 4.2. In this example noise in the raw data has led to a number of slightly different trails leading to the same grid square (bottom left). In general these would be considered the same and this bottom left square would be considered the best prediction, which is what the proposed approach provides. In contrast, traditional approaches consider all the other trails separately and propose the center right grid square.

The motivation for the proposed approach over the traditional approach can be seen from two angles. Firstly, depending on how the raw GPS logs are decomposed into trails, the error shown in figure 4.1 can be reduced by restricting the encoding of trails to not include loops. As discussed in 2.1.1 there are many different ways to decompose raw GPS logs into trails. Falling into two major categories, these can be seen as either heuristic or knowledge-based. In both cases, however, the goal is typically to compose trails between destinations, with the latter clearly providing a less noisy realization of this goal. As such loops are not the primary objective in the applications typically motivated. Rarely modelled, such occurrences are typically the result of noise within the trail decomposition mechanism, with none occurring under many knowledge-based decomposition schemes. Secondly, the relaxed approach allows for the application of Bayes' theorem to approximate the probability of a given spatial region occurring in the future given an arbitrary set of spatial regions occurring in the past. I.e. using the informal notation:

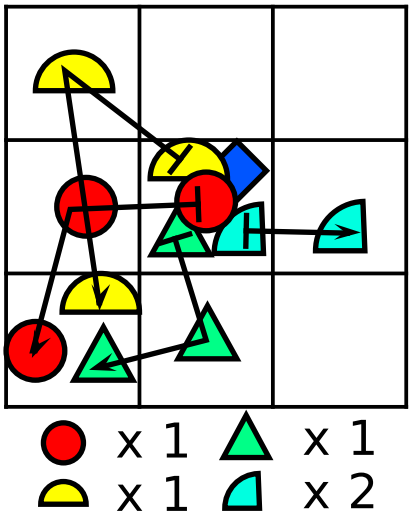


Figure 4.2: An example showing a case where the proposed approach will outperform traditional Markovian approaches.

$$\forall s_i, s_v \in S, P(s_{i \text{ in future}} | s_{v_1} \wedge \ldots \wedge s_{v_p \text{ in past}}) \tag{4.3}$$

The above formulation is the foundation of the proposed approach.

4.2.1 A Naive Bayes Model

The relaxed encoding discussed in the previous section provides the ability to apply Bayes' rule to approximate probabilities for future regions considering full variable history lengths. The benefits of using full variable history lengths has been detailed previously in the beginning of this thesis in chapter 2, section 2.1.3. The approximation is made by the following assumption:

Assumption 1. *Given the current state, the probability that a specific, alternative, state occurred in the past is only is dependent on the current state.*

Clearly such an assumption is not true as the probability is also generally dependent on the other past states that were part of the trail taken up to that point. However, naive Bayes' systems have been shown to work surprisingly well in real world scenarios (e.g. [24, 151]) and its application to the GPS domain is later validated empirically in section 4.4. In any case it seems intuitive that such a system will perform better than a first order Markov chain due to the use of more historical information. A simple example highlighting this is shown in figure 4.3, where two different paths would be taken

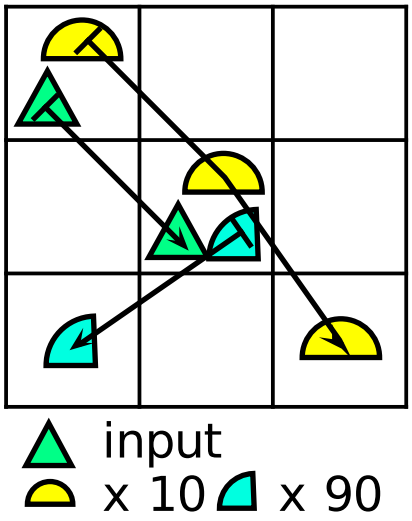


Figure 4.3: An example showing the importance of the input's history

depending on if the first piece of the input (represented by triangles) is considered.

Formally Bayes' rule is defined as:

$$P(b|a) = \frac{P(a|b)P(b)}{P(a)} \tag{4.4}$$

If only the probability that a spatial region occurred given the presence of another spatial region in a trail is considered (i.e. if it was not stipulated that the spatial region being reasoned about must be in the future of the given spatial region) then we could simply apply Bayes' rule and the corresponding assumption of conditional independence to equation 4.3. Let $E = \mathsf{e}_1, \ldots \mathsf{e}_{|E|}$ $(e \in S)$ denote the prediction input which consists of a set of observed regions (evidence). The resulting probability of an arbitrary region $\mathsf{a} \in S$ is then:

$$\begin{aligned} P(\mathsf{a}|\mathsf{e}_1 \wedge \ldots \wedge \mathsf{e}_{|E|}) &= \frac{P(\mathsf{e}_1 \wedge \ldots \wedge \mathsf{e}_{|E|}|\mathsf{a})P(\mathsf{a})}{P(\mathsf{e}_1 \wedge \ldots \wedge \mathsf{e}_{|E|})} && \text{Applying Bayes' Rule} \\ &= \frac{P(a)\prod_{i=1}^{|E|} P(\mathsf{e}_i|\mathsf{a})}{P(\mathsf{e}_1 \wedge \ldots \wedge \mathsf{e}_{|E|})} && \text{Assuming conditional independence} \\ &= \gamma P(\mathsf{a}) \prod_{i=1}^{|E|} P(\mathsf{e}_i|\mathsf{a}) \end{aligned} \tag{4.5}$$

In this case $\gamma = \frac{1}{P(\mathsf{e}_1 \wedge \ldots \wedge \mathsf{e}_{|E|})}$ and is a fixed constant for each prediction, over all possible predictions, since it refers to only the route taken so far. Being a constant it therefore does not effect the relative ranking of possible predictions and as such does not need to be calculated.

The above formulation is widely used and known as the naive Bayes model. In the case of movement prediction, however, an alteration is proposed in order to model the probability of a spatial region occurring in the future given a set of spatial regions that have occurred in the past as informally notated in equation 4.3. The formulation is almost identical although the derivation must start from a more primitive starting point, the definition of probability with respect to future and past.

Let the probability of a spatial region occurring in the future be denoted by a superscript f and the probability of a spatial region occurring in the past be denoted by a superscript p. Additionally define the function $count(\mathsf{a}_i \wedge \ldots \wedge \mathsf{a}_j)$ as the number of trails in which all spatial regions $\mathsf{a}_i \ldots \mathsf{a}_j$ occur. Finally define the function $count(\mathsf{a}_i \wedge \ldots \wedge \mathsf{a}_j \xrightarrow{before} \mathsf{a}_k)$ as the number of trails in which all spatial regions $\mathsf{a}_i \ldots \mathsf{a}_k$ occur *before* region a_k.

Based on the above notation define the probability of spatial region $\mathsf{a} \in S$ occurring in the future of an arbitrary number of other spatial regions $E = \{\mathsf{e}_1, \ldots, \mathsf{e}_{|E|}\}$ and the probability of e_1 through $\mathsf{e}_{|E|}$ occurring in the past given a as equations 4.6 and 4.7 respectively:

$$P(\mathsf{a}^f | \mathsf{e}_1 \wedge \ldots \wedge \mathsf{e}_{|E|}) = \frac{count(\mathsf{e}_1 \wedge \ldots \wedge \mathsf{e}_{|E|} \xrightarrow{before} \mathsf{a})}{count(\mathsf{e}_1 \wedge \ldots \wedge \mathsf{e}_{|E|})} \tag{4.6}$$

$$P(\mathsf{e}_1^p \wedge \ldots \wedge \mathsf{e}_{|E|}^p | \mathsf{a}) = \frac{count(\mathsf{e}_1^p \wedge \ldots \wedge \mathsf{e}_{|E|}^p \xrightarrow{before} \mathsf{a})}{count(\mathsf{a})} \tag{4.7}$$

Using 4.6 and 4.7:

$$P(\mathsf{a}^f | \mathsf{e}_1 \wedge \ldots \wedge \mathsf{e}_{|E|}) = \frac{P(\mathsf{e}_1^p \wedge \ldots \wedge \mathsf{e}_{|E|}^p | \mathsf{a}) count(\mathsf{a})}{count(\mathsf{e}_1 \wedge \ldots \wedge \mathsf{e}_{|E|})} \tag{4.8}$$

As in the naive Bayes' formulation note that for a given prediction $\mathsf{e}_1 \wedge \ldots \wedge \mathsf{e}_{|E|}$ is fixed and only the ordering of the potential spatial regions is required. As such the constant $count(\mathsf{e}_1 \wedge \ldots \wedge \mathsf{e}_{|E|})^{-1}$ can be dropped.

Next assumption 1 is applied, which, in the above notation can be formally written as:

$$P(\mathsf{e}_1^p \ldots \mathsf{e}_{|E|}^p | \mathsf{a}) = P(\mathsf{e}_1^p | \mathsf{a}) \ldots P(\mathsf{e}_{|E|}^p | \mathsf{a}) \tag{4.9}$$

Substituting the assumption into 4.8 leads to equation 4.10, the equivalent to the naive Bayes' formulation.

$$P(\mathsf{a}^f | \mathsf{e}_1 \wedge \ldots \wedge \mathsf{e}_{|E|}) \propto P(\mathsf{e}_1^p | \mathsf{a}) \ldots P(\mathsf{e}_{|E|}^p | \mathsf{a}) count(\mathsf{a}) \tag{4.10}$$

The formulation allows the reasoning over all spatial regions given an arbitrary history of spatial regions by just keeping a matrix of size $n \times n$ where n is the number of cells the spatial area of interest was quantized into. This is the same space requirement as the Markov model in theory. However, a first order Markov model (assuming movement trails with no gaps after spatial quantization) generally only ends up providing probabilities for each location conditional on the surrounding cells. In contrast the proposed approach will generally provide probabilities for each location conditional on those that interact with that location, which tends to be a larger set. Therefore while both approaches have the same theoretical space requirement the standard Markov model tends to result in a sparser matrix of probabilities, and hence a smaller memory footprint.

In the remainder of this chapter this proposed basic naive Bayes' model will be called the *relaxed model*. In this basic form the model does not perform overly well, however, an additional two extensions, encoding direction and negative information, make the model much more accurate and are discussed in the next two sections.

4.2.2 Encoding direction

The basic model presented in equation 4.10 in the preceding section suffers from a number of potential points of error. In this section the error introduced by neglecting to encode direction is considered. In a full history model direction is encoded implicitly. If the history is $a \rightarrow b \rightarrow c$ then it is obvious that from spatial region a movement was then in the direction to b and then in the direction towards c. Within the naive Bayes' model, however, only the probability that a region occurred in the future given a region is kept and hence no directional information can be inferred. As such the model can not capture behaviour that only differs in direction. A simple example is presented in figure 4.4. Later in section 4.4 empirical results are presented which highlight the importance of direction.

Direction is added to the model presented in equation 4.10 at the fine grain level of individual spatial area transitions. This is achieved by encoding spatial area transitions rather than the spatial areas themselves. This is similar to the encoding proposed in [139] where cell edges were enumerated, although the algorithmic context is different. However, in this case the additional encoding of diagonal movements is proposed, leading to each previous spatial area being represented by eight probabilities rather than one. A directional encoding scheme is used rather than edges since the directional encoding

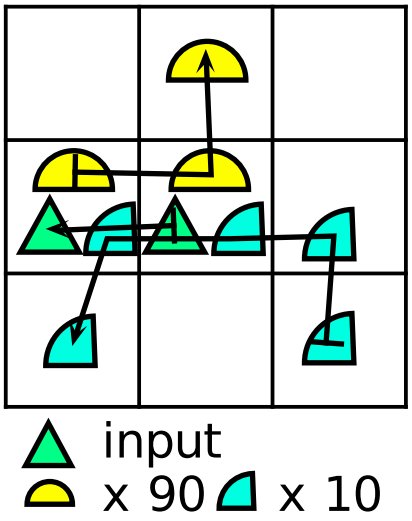


Figure 4.4: An example showing the importance of direction in prediction from historical data.

can handle the case where a cell is skipped. In this an *up movement* is only encoded if the location being transitioned to is directly above the previous location. Any slight deviation to the left or right changes the encoding to up-left or up-right. Encodings of down, left and right are done in a similar fashion with respect to movement involving skipped locations. This can lead to a disproportional amount of diagonal movement encodings. Alternative encoding methods could be used to alleviate this, however, in this initial proposal the simpler approach is taken. Therefore, again under the notation of section 4.2, the probabilities being modelled are altered from:

$$\forall s_i, s_v \in S, P(s_{i \text{ in future}} | s_{v \text{ in past}})$$

to:

$$\forall s_i, s_v \in S, \forall d_j, d_w \in W, P(\vec{s}_{i \text{ in future}}^{d_j} | \vec{s}_{v \text{ in past}}^{d_w}) \tag{4.11}$$

Where $\vec{s}_i^j$ represents spatial area s_i which was reached by a transition in direction d_j and W is the set of all transition directions. Figure 4.5 graphically shows the encoding process on a small example. As is expected, such an approach comes at a cost. Storage requirements increase from the previous $n \times n$ probabilities to an upper limit of $8(n \times n)$ probabilities. It is of note, however, that when utilizing sparse matrices, the actual storage space required is typically significantly less in practice due to the relative sparsity and repetition of movement patterns.

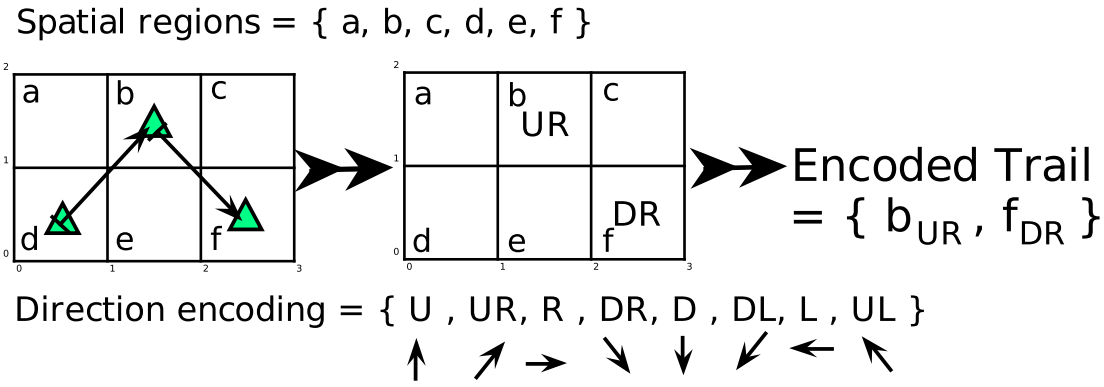
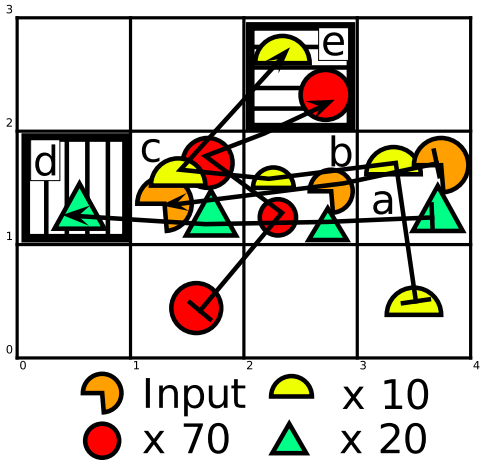


Figure 4.5: A graphical example of the proposed directional encoding scheme.



Under directional encoding we have [b, L] and [c,L] as the input's history
Consider potential prediction regions e and d.
Therefore, we generate:
P([e, UR] | [b,L] ^ [c,L]) = 0.8 * 0.7 * 0.1 = 0.056
P([d, L] | [b,L] ^ [c,L]) = 0.2 * 0.2 * 0.2 = 0.008
and select region e due to the higher score of 0.056
However, d is more likely given the full history of the input. The model has been contaminated by a composite effect of a unrelated historic trail.

Figure 4.6: An example showing a case where the less intuitive prediction would be made without the use of negative information.

4.2.3 Encoding negative information

The augmented model presented informally in equation 4.11 allows the model to incorporate variable length input and encodes direction. However, since each spatial region and direction combination is modelled independently (see assumption 1), each spatial region under consideration for prediction gets an increased weighting each time any of the input's history occurs with each region under consideration in a historic trail. As such unrelated trails that share only a small portion of the input and the spatial region under consideration can artificially increase the prediction score for that region. An example highlighting this problem is shown in figure 4.6.

In order to address the problem the use of negative information, which can be inferred, is proposed. Specifically note that if a movement to a spatial region from a certain direction occurred, then it is also known that movements to this region from other directions

did not. This information can then be utilized in the model as negative evidence. The model is shown below.

Let $W = d_1, \ldots, d_8$ be the set of directions that a spatial area $s \in S$ is reached via. Given an input set of spatial areas and their transition information $E = \bar{\mathsf{e}}_1^{d_p}, \ldots, \bar{\mathsf{e}}_{|E|}^{d_q}$ Then each $\bar{\mathsf{e}}_i^{d_p}$ is expanded via the function $eNeg(\bar{\mathsf{e}}_i^{d_p})$ to include negative information, with the function defined as:

$$eNeg(\bar{\mathsf{e}}_i^{d_p}) = \bar{\mathsf{e}}_i^{d_p} \bigwedge_{\forall d_q \in D, d_q \neq d_p} \neg \bar{\mathsf{e}}_i^{d_q} \tag{4.12}$$

The probability of each potential future location is then calculated as normal using the expansion:

$$P(\mathsf{a}|eNeg(\bar{\mathsf{e}}_1^{d_p}), \ldots, eNeg(\bar{\mathsf{e}}_{|E|}^{d_q})) \tag{4.13}$$

The inclusion of the negative information penalizes any historic trails that enter the input's history in a way that is not consistent with the history, thereby preventing the contamination shown in figure 4.6. It is of note that this aspect of the model enforces the constraints locally and therefore relies on both the input and the historic trails being continuous, in other words for each point in the quantized trail to be adjacent to the preceding and proceeding points, to work as designed. Unfortunately this is not guaranteed in practice. In reality noise in the GPS sensors can introduce gaps in a trail, although this problem is rare if the quantization level of the spatial area is coarse enough with respect to the GPS sampling rate. Optionally this can be alleviated by interpolation. Interpolation, however, is prone to introducing additional errors in the case where a GPS point is erroneously recorded as a long way off from the actual location. Specifically numerous additional erroneous points are interpolated and used as the input to the prediction algorithm, or as assumed correct historic trails[2].

The same model can also be used to enforce the importance of the start point by including negative information enforcing that the initial region did not come from any direction. Note that this is a modelling decision[3]. A theoretical motivation for such a constraint

[2]While not formally examined in this work pilot studies on a reduced subset of the data led to poorer results. While the study was too small to provide sufficient evidence that this is the case in general a full formal evaluation would make interesting future work.

[3]This is examined in more depth in chapter 6 where an encoding emphasizing the start point is shown to perform relatively well.

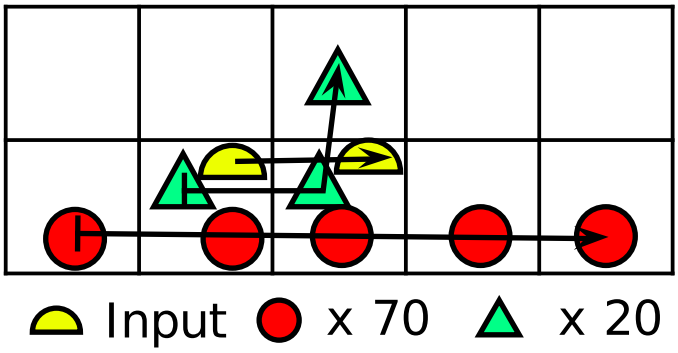


Figure 4.7: Example showing the importance of the start location of a trail.

is the realization that long common corridors of movement may have trails starting at various points and it is these points that act to discriminate the final route. For example, consider a major road within a city. Many cars (particularly over time) may use the road from its start to finish making the full length of the major road the most likely prediction for observations along the route. However, people who live along the road may typically go to local shops or other local locations, hence the general prediction would be incorrect in the majority of these cases. A visualization of the example is shown in figure 4.7. In the predictors evaluated in this chapter using negative information the choice has been made to enforce the importance of the start point in the encoding.

In terms of implementation, the use of negative information requires the ability to reason about the probability of something not occurring in the past given a future state. Note this is due to the implementation of equation 4.13 as the Bayes approximation (equation 4.10). Therefore the implementation requires the reasoning, and hence calculation, of the following probabilities (in the notation used in section 4.2.1):

$$P(\neg \mathsf{a}^p | \mathsf{k}^f) = P(\neg \mathsf{a} | \mathsf{k}) + P(\mathsf{a}^f | \mathsf{k}^p) \tag{4.14}$$

where a and k are arbitrary symbols in S.

However, in the context of movement prediction $P(\mathsf{a}^f|\mathsf{k}^p)$, the probability of a being in the future given k is in the past is not a desirable contributor to the probability that k is in the future and hence we require only $P(\neg \mathsf{a}|\mathsf{k})$. Unfortunately $P(\neg \mathsf{a}|\mathsf{k})$ can not be calculated directly from previously stored information and so it must be calculated and stored separately. However, in practice $P(\mathsf{a}|\mathsf{k})$ is calculated and stored as it is sparse and $P(\neg \mathsf{a}|\mathsf{k})$ is calculated at runtime via:

$$P(\neg \mathsf{a}|\mathsf{k}) = 1 - P(\mathsf{a}|\mathsf{k}) \tag{4.15}$$

Unfortunately this approximately doubles the space requirement of the method. It is of note that this can be avoided by giving up the encoding of future and past. In this case only the second set of probabilities needs to be maintained and they can be used for both the positive and negative probabilities by equation 4.15 and the traditional naive Bayes' model detailed in equation 4.5. An argument for this approach is that under the directional encoding the need to explicitly keep track of when regions occur in the future is less marked as the direction of travel implicitly encodes this information in most cases.

4.3 Prediction mechanisms

The model described in the previous sections enables the generation a set of probabilities over all spatial region and direction combinations. However, in order to be utilized as a prediction system a prediction mechanism must be implemented and a number of issues addressed. Specifically:

1. *Prediction beyond the next time step*
   While most previous work has addressed simply the next time step, this work seeks to predict routes of arbitrary length. A naive approach is to try and enumerate all possible combinations of spatial regions that could contribute to the prediction. With order being important, however, this quickly leads to an intractable number of combinations when predicting a large number of time steps into the future.

2. *The exact number of time steps to predict for*
   Unlike techniques that match the input to a historic trail and simply use the remainder of the historic trail as the prediction, fixed length Markov models and consequently the proposed method requires a stopping condition.

3. *How to proceed when the method does not produce any output*
   Typically this occurs when the inputs have not been seen together with any of the potential prediction regions.

4.3.1 Predicting beyond the next time step

Markov models provide a probability of a region being the next step and in contrast the proposed model provides a probability of a region being visited in the future. In

either case probabilities are defined over all locations and the probabilities are valid for the next step/future for the given history. As such, predicting a route requires further processing. In the case of traditional Markov Models, where the probabilities are next step probabilities, the obvious solution is to consider all locations and select the one with the highest probability. This then becomes the next point in the prediction. The prediction can then replace the route taken so far and the next step predicted recursively. Depending on the number of locations, considering all locations may be computationally restrictive. This can be alleviated by considering routes to be somewhat continuous and only considering locations within a certain spatial region. Since computational complexity is of concern in this context (otherwise more complex predictors available within the literature could be used) this is implemented in the algorithms used in this chapter, with the spatial region considered being a ten metre radius from the boundary of the spatial area under consideration.

In contrast to the traditional models which provide next step probabilities, the proposed model provides probabilities of future occupancy and therefore a wider range of options exist for predicting routes. In order to ensure the approach remains computationally inexpensive only recursive options, which represent heuristic approaches, are considered. While route selection based on a globally optimal criteria is possible it would be considerably more computationally expensive. A first recursive option is to mimic the approach taken with traditional Markov Models, considering the more general future occupancy probabilities as next step probabilities. A second is to select the most probable path by selecting the most probable location within a spatial radius recursively. Unlike the first approach the newly predicted point is then not used as part of the history for the next iteration. This is a valid approach since the probabilities are future occupancies. The approach prevents the choice of later locations being as effected by the selection of a given location, as it is not then used to dictate the next set of probabilities from which the next location is selected. Due to this the second approach is implemented in the predictor and its variants presented and evaluated in thesis. While interesting, the investigation into variants of the route selection mechanisms are left for future work. It is of note that in both the traditional Markov Model and the proposed approach the examination of all locations in a user-defined spatial radius addresses the issue of non-continuous trails in a parametrized way. In the methods evaluated in this chapter this parameter is consistently set to ten metres in both the proposed naive Bayes' based predictors the Markov Model based predictors.

4.3.2 The exact number of time steps to predict for

All models discussed in this chapter provide probabilities over locations. As such the end of the prediction can be hard to ascertain[4]. A naive approach is to predict until all possible regions for prediction have zero probability. While initially this may seem sensible, it is clear that such an approach will favour longer predictions. A concise example is shown in figure 4.8. An alternative approach utilized in [202] when using a Markov model as baseline was to predict until the average length of historic trails. Clearly, however, such a one size fits all approach is not ideal and, depending on how representative the average actually is, can produce predictions of incorrect length more often than not. In this work this problem is not addressed directly. Rather it is noted that in many cases the use of an application-specific threshold value can be employed to stop the recursive prediction algorithm if computational simplicity is desired. Since the system generates probabilities, such a threshold has a fixed meaning. If the destination is of utmost concern then other models may be better suited for the particular problem, with many researchers investigating the slightly different problem of significant place detection [11, 119] and destination prediction [147**?**]. Alternatively if both destination and route prediction are desired more complex models such as those detailed in section 2.1.3 and the method proposed in chapter 6 are likely better alternatives.

In evaluation the view is taken that route prediction is the primary concern over destination prediction and therefore the naive stopping point - zero probability for any potential next step regions is used. As discussed this typically results in over predictions. When comparing the correct trail against the prediction, extra length is then not penalized, as discussed in chapter 3.

4.3.3 Possibility of no predictions

The models considered in this chapter require the spatial area of interest to be quantized into discrete symbols, against which probabilities are assigned. Since the probabilities are built from historic data there is always the possibility that the input sequence has not been seen before. A basic approach is to consider these inputs to have zero probabilities within the model. This is discussed as the closed world model in [**?**] where the authors

[4]This is in contrast to pattern matching approaches discussed in chapter 2 where the end of the prediction is the end of this historic trail matched.

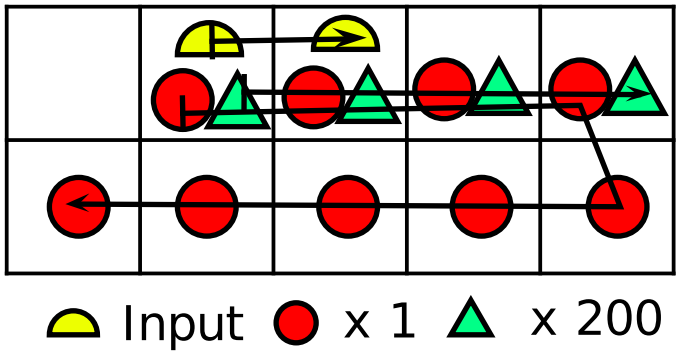


Figure 4.8: An example showing how a strategy of recursive prediction with a stopping condition of zero probability in surrounding cells leads to an unintuitive over prediction (the circles).

argue that such an approach is naive since people can visit locations they haven't seen, noting that this is particularly prevalent in the early stages of observing a driver. They go on to argue that a fixed probability should not be used either, highlighting this with the example of the center of a lake as a place where the probability should be zero. Instead another assumption is proposed based on their specific application, driving, deriving a function to determine approximate probabilities. When considering aggregated logs the issues raised in [**?**] lessen - the use of large collections of logs aggregated over multiple people helps to saturate the model. Due to this, setting non-zero probabilities to unseen locations based on domain knowledge is not dealt with this thesis, although such work may help to increase prediction accuracy in a small number of cases. In contrast to the driving domain, however, predicting pedestrian routes does not allow the correction of raw GPS points to roads using external knowledge and as such a new problem, the presence of noisy inputs leading to paths appearing different that in reality are identical. As such the noise in a GPS reading can artificially create an input that has not been seen before. Particularly it is a problem in the model proposed in section 4.2 since if any part of the input's history is unseen, with respect to all of the potential next step regions, no prediction will be made. As with all problems of noise, employing a coarser level of quantization can help, however, clearly this is not a desired solution. It is of note that this problem is not limited to models using the Markov property, but occurs in all models to some degree. For instance, in [137] where a variable length model is used, the authors utilize a distance function to calculate the historic trails closest to the input history in order to deal with the problem where the input does not exactly match a historic trail. In [139] three different strategies are

evaluated and in [182] a fallback strategy was implemented when evaluating order k Markov models. The fallback strategy simply involves trying $k-1$ order Markov models repeatedly until either a prediction is made or $k = 0$. In the context of prediction based on an input's history, this is equivalent to reducing the input's history one at a time by removing the oldest observed spatial region and running the algorithm again. The approach showed reasonable prediction accuracy in the evaluation [182] and it is this strategy that is adopted in this work. The approach also represents the computationally cheapest approach, in line with the previous arguments that Markovian algorithms must be computationally cheap to be useful. Note that the alternative of finding the *closest* match as per [137] is more complex in the case of Markovian based models since the independence assumption removes the order information. Therefore, without retaining all historic instances separately, all spatial regions near missing regions would have to considered, where *near* is a user-defined parameter. Even if this was implemented it is possible that it is noise that has been matched and it would have been better to simply remove the region in question from the input's history. Since it is unknown when, or if, this kind of approach would be beneficial it is not considered.

4.4 Experimental Results

In order to evaluate the proposed approach and the proposed encodings, a comparison of four variants of the sequence-relaxed Naive Bayes' model, a variant of the traditional first order Markov model, along with three baseline methods was undertaken under two different levels of spatial quantization, resulting in 16 different model variants.

4.4.1 Baseline methods

In order to compare the new approach three baseline approaches are included in the evaluation. The first is the basic first order Markov Model. The second and third are computationally complex predictors based on state-of-the-art matching via prefix trees using custom distance functions as discussed in chapter 2, section 2.1.3.

The first of these complex predictors is the predictor proposed in [137]. The predictor fuses a distance function and a measure of support to determine the most similar historic trail to the input. More specifically, the approach builds a prefix tree from the historic

observations. Each node other than the root contains:

1. a symbol denoting the region it represents
2. the number of times the trail up to that node (from the root) was seen in the historic set (the support)
3. a list of child nodes

Each edge is additionally annotated with a range, which represents the minimum and maximum transition times between the parent and child nodes that the edge connects.

Prediction is performed by selecting the *best matching* branch from the root that matches the input sequentially. Once the input is depleted, the remaining branches from the final matched node become the potential predictions. Support values for each branch are then used to determine the most likely prediction[5]. This is a slight deviation from the originally proposed algorithm which only considers the next step prediction, and therefore simply considers the children not the whole branch under the last matched node.

The best matching branch can be defined in a number of ways. However, the sum aggregation function is shown to perform the best in [137], and so the sum aggregation function is used in the baseline implementation.

Each pairwise comparison is scored via function 4.16. The function uses three user-defined parameters, tw (temporal weighting), sw (spatial weighting) and tt (temporal tolerance). The spatial and temporal weightings range from zero and one and define the relative weight of the spatial and temporal penalties respectively. The temporal tolerance defines the maximum the match can be out by with respect to the transition time before it is not considered to be a match even when the spatial symbols are an exact match. The function takes the two symbols (e,f) being compared. e represents the input symbol which has an associated transition time (the time it took to transition to the symbol from the previous symbol), accessed by the function $time(\cdot)$. f represents the symbol being matched which is part of a branch in the tree. f has the three attributes defined above, a support value (accessed by $sup(\cdot)$), a minimum transition time (accessed

[5]Note that since that only the branches that result from the history matching the input up to this point are considered the branch with the highest support is also the branch with the highest conditional probability (also known as confidence - see chapter 5).

by $minTime(\cdot)$) and a maximum transition time (accessed by $maxTime(\cdot)$).

$$score(\mathsf{e},\mathsf{f}) = \begin{cases} sup(\mathsf{f}) & \text{if } \mathsf{e} == \mathsf{f} \text{ and } in(\mathsf{e},\mathsf{f}) \\ \frac{sup(\mathsf{f})}{tw \times d_t(\mathsf{e},\mathsf{f})} & \text{if } \mathsf{e} == \mathsf{f} \text{ and } \neg in(\mathsf{e},\mathsf{f}) \text{ and } d_t(\mathsf{e},\mathsf{f}) < tt \\ \frac{sup(\mathsf{f})}{(tw \times d_t(\mathsf{e},\mathsf{f})) + (sw \times |\mathsf{e}-\mathsf{f}|)} & \text{otherwise} \end{cases} \quad (4.16)$$

where:

$$in(\mathsf{e},\mathsf{f}) = \begin{cases} true & \text{if } minTime(\mathsf{f}) \leq time(\mathsf{e}) \leq maxTime(\mathsf{f}) \\ false & \text{otherwise} \end{cases}$$

$$d_t(\mathsf{e},\mathsf{f}) = \max(minTime(\mathsf{f}) - time(\mathsf{e}), time(\mathsf{e}) - maxTime(\mathsf{f}))$$

Intuitively the score (and hence the overall aggregated score) is a fusion of the support values and the spatial and temporal distance functions. It is of note that the authors also employ thresholds to return *no prediction* when the match is to far out either spatially or temporally. This prevents predictions that are vastly incorrect from skewing the reported averages. Since this is dealt with under the evaluation metrics proposed in chapter 3 this is not required. It is of note that the method requires all paths of the tree to be considered in order to find the globally maximal branch.

The predictor requires the definition of the three parameters, sw, tw and tt. Of these the temporal threshold parameter (tt) is fixed to six seconds since this is the value used to convert the continuous GPS data streams into trails. Good values for the weighting parameters, however, are unknown and therefore these parameters are chosen experimentally using the same data set as in the evaluation. Generally considered bad practice, this potentially enables the predictor to perform better than would be expected in the real world where the predictor would need to have the parameters set from an independent data set first. However, considering this method is only used as a baseline this is of limited concern. The parameter values investigated were $0.0, 0.25, 0.5, 0.75, 1.0$ for temporal weighting and $0.25, 0.5, 0.75, 1.0$ for spatial weighting under the conditions that are later used in the evaluation. Spatial weighting was not set to 0.0 since this results in almost all historic trails matching making the method intractable. The investigations were carried out for both levels of quantization (5 and 10 metres). The corresponding parameter plots are shown in figures 4.9 and 4.10 respectively.

Based on these results the spatial weights are set to 0.25 and 1.0 and the temporal

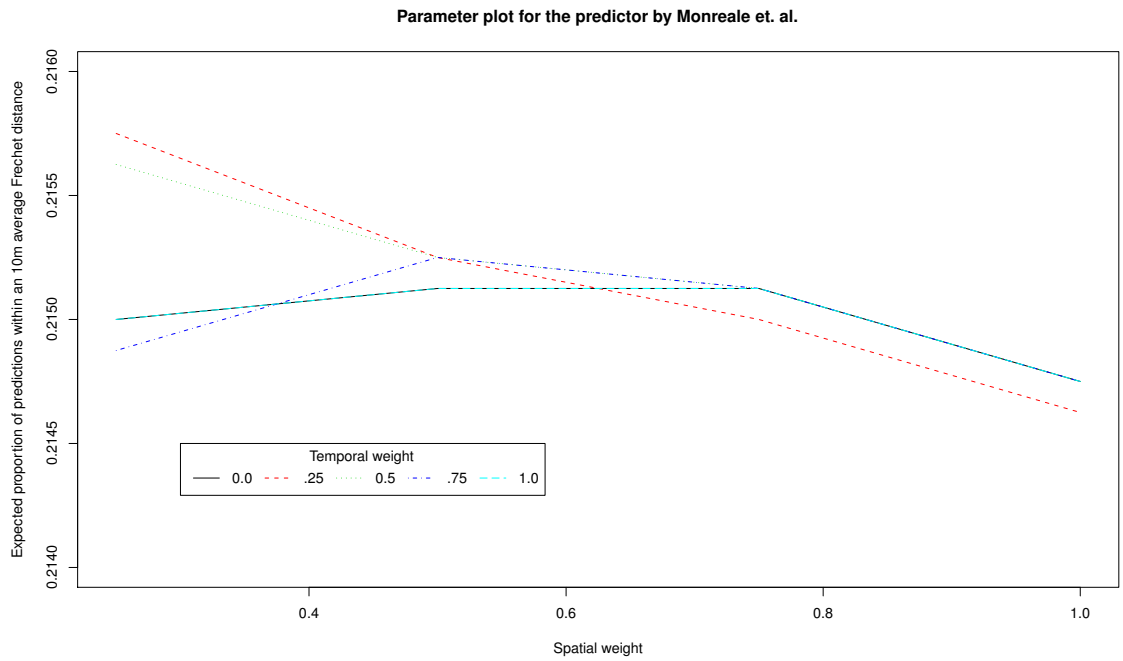


Figure 4.9: Graph showing the mean proportion of routes predicted within an average Fréchet distance of 10 metres for the predictor from [137] using different spatial and temporal weights, when 5 metre quantization has been performed.

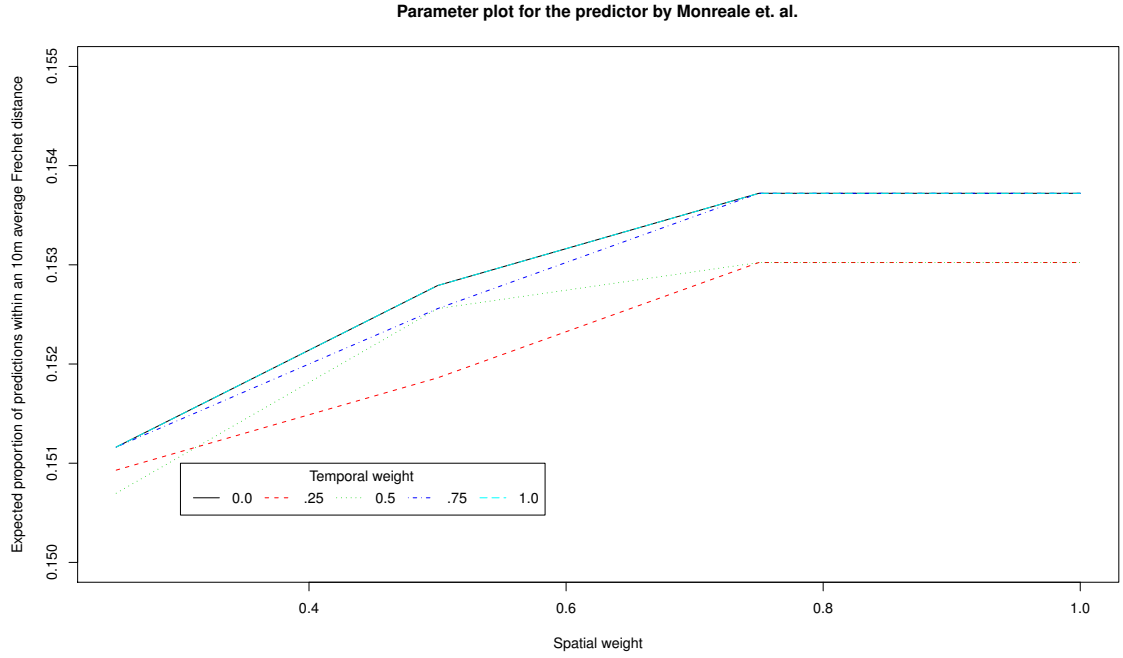


Figure 4.10: Graph showing the mean proportion of routes predicted within an average Fréchet distance of 10 metres for the predictor from [137] using different spatial and temporal weights, when 10 metre quantization has been performed.

weights set to 0.25 and 0.0 respectively for 5 and 10 metre quantization.

The second complex predictor used as a baseline is one that was developed during the implementation of the predictor from [137], aimed at removing the parametrization of the approach. The new predictor uses the same data structure but simplifies the approach removing all parameters bringing the approach back to a simple context pattern matching approach. The approach is very similar to that proposed by [139] where a distance function selects candidate patterns purely on spatial properties. If multiple patterns have the same score then the probability of the remainder of the pattern, given the matched section (conditional probability), is used to select the returned prediction[6]. This predictor is used as the second of the complex predictor baselines, since in testing the approached showed superior results to any parameter combination of the predictor from [137]. While this is potentially a contribution in its own right, the predictor requires[7] distance calculations between the input and each frequent pattern (historic trail). This was considered more computationally expensive than desired and other methods developed. The predictor is therefore used only as a more stringent baseline than the approach from [137], representing a straight forward combination of the concepts from the predictors proposed in [137, 139] with superior prediction accuracy.

Specifically the approach simply redefines the score function from [137] (equation 4.16) to:

$$score(\mathsf{e}, \mathsf{f}) = |\mathsf{e}, \mathsf{f}| \tag{4.17}$$

The lowest score is then taken rather that highest. The support[8] is used to select from multiple candidate predictions if multiple candidates exist.

4.4.2 Methods evaluated

The four variants of the sequence relaxed Naive Bayes' model evaluated in the study seek to investigate the effect of the various proposed modifications to the base algorithm motivated in sections 4.2.2 and 4.2.3. The three baseline variants implemented were

[6]It is possible, but uncommon, that there are more than one pattern with the same matching score and conditional probability. In evaluating the predictor, if this occurs, the worst performing pattern is used.

[7]In the worst case. Pruning the search space is possible if close matches are found early. Additionally greedy heuristics could be used, however, these may not find the best overall match and are not considered either here or in [137].

[8]Again, also equivalent to conditional probability.

those discussed in the previous section, the standard first order Markov model, the predictor proposed in [137] and the alternative complex predictor. Additionally a fifth model, a standard first order Markov Model was implemented using the directional encoding, in order to better isolate and evaluate the effect of the encoding. We note that it is not possible to utilize negative information in a first order Markov Model and hence such a variant was not considered. First order Markov Models were chosen since they utilize the same recorded information representing a fair comparison with respect to space complexity compared to two of the variants. The impact of spatial quantization on all models was investigated via two levels of spatial quantization. The first was using five metre grid quantization which approximately represents the granularity that the data set was recorded at, with GPS points being recorded every five seconds and only when the participant had moved more than five metres. The second was a more coarse grain grid of ten metres.

The evaluated predictors and the acronyms used throughout the rest of the chapter are detailed below. All algorithms denoted NB are variants of the predictor proposed in this chapter.

NB The relaxed naive Bayes' model, including order distinction (equation 4.10), directional encoding and utilizing negative information

NB-D NB without directional encoding

NB-O NB without order distinction (model built using equation 4.5)

NB-N NB without utilizing negative information

MM The standard first order Markov Model

WN An implementation of the algorithm in [137] with parameters set as described in section 4.4.1.

SEQ The second complex predictor developed as a baseline as described in section 4.4.1.

MM-V MM The standard first order Markov Model using the direction encoding scheme.

4.4.3 Evaluation methodology

The experimental methodology follows the recommendations made in chapter 3. Specifically, the truncated average discrete Fréchet distance is used as the test statistic for a single prediction, and repeated 10-fold cross-validation is used for the testing methodology, with 10 repeats. The results are then graphed, displaying the expected generalized proportion of correct predictions with respect to a varying relevance parameter, side-by-side histograms plotted of the individual predictions and side-by-side box plots provided displaying the aggregated results. Since spatial quantization has been performed, all units reflect the quantized scale. Therefore using a quantization of 10 metres predictions within one unit are actually predictions within 10 metres. For statistical analysis paired t-tests using a conservative estimate of the variance (specifically the OET method motivated in section 3.2.3 with $M = 7$) was used. Since multiple comparisons were performed, p-values were adjusted according to Holm's procedure. Holm's procedure was used in contrast to the Bergmann-Hommel procedure since it is easily available in the R package *muToss* [140]. The use of the Holm over the Bergmann-Hommel procedure further contributes to the conservative nature of the statistical procedure, meaning any reported significant results are at least significant at the reported level, and potentially significant at lower levels.

The dataset used in the evaluation was the *D-SCENT* dataset, as discussed in chapter 1, section 1.2.5. Covering $80,000m^2$ the data set involves 60 participants. Each individuals path was broken into trails. This was achieved by splitting an individuals GPS stream when a gap of six seconds or more occurred. Trails were then only kept if they were at least 20 metres in length and had at least five points. In total 1031 individual trails were created with an average length of 128.7 metres. Following trail extraction quantization was performed. After quantization only trails with length of at least 5 were kept to ensure that each trail could be split into input and ground truth segments in a meaningful way. For evaluation historic trails were split into two at a randomly selected split point with the first section of the trail used as input and the second used as the prediction ground truth. A minimum length of two and three for the input and ground truth was enforced respectively. This reflects the desire to evaluate route prediction and not just next step predictions. Quantization via a 10×10 metre grid lead to a total of 438 movement trails being extracted, with an average a trail consisting of 7.881 spatial regions with a standard deviation of 3.290. When quantized by a 5×5 metre grid, 805 movement trails

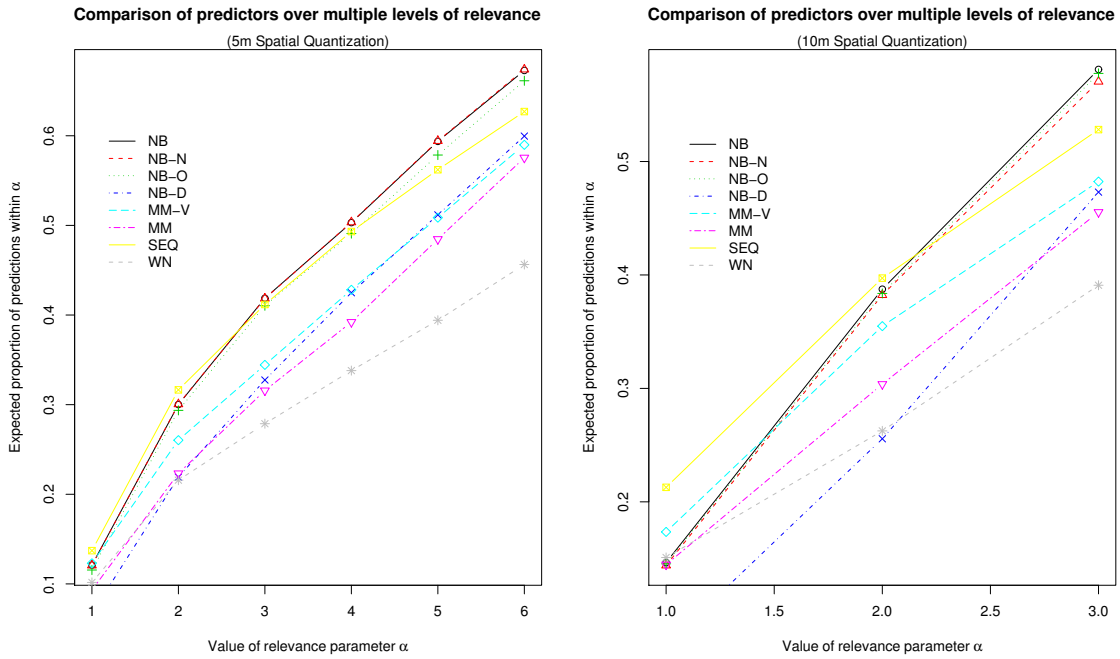


Figure 4.11: Graph showing the prediction accuracy of the predictors for different levels of the relevance parameter α for 5 and 10 metre spatial quantization. The parameter α is in units based on the spatial quantization.

were extracted. An average trail consisted of 10.399 spatial regions with a standard deviation of 5.947.

4.4.4 Results

Prediction accuracy was evaluated using ten times repeated 10-fold cross validation, using the truncated average discrete Fréchet distance as the test statistic for a single prediction. The error function (as motivated in section 3.1.9) then converts the Fréchet distances from each fold into a proportion based on a user defined level of an acceptable distance (α). A range of α parameter options are examined with respect to both 5 and 10 metre spatial quantization and a line graph plotting the parameter vs. prediction accuracy is shown in figure 4.11. Prediction accuracy refers to the average proportion across all folds per predictor. The x-axis shows the value of the relevance parameter in units determined by the level of spatial quantization. In both cases the range of the values is up to 30 metres. In the case of $5m$ spatial quantization this means α is set to 1,2,3,4,5 and 6. In the case of $10m$ spatial quantization α is set to 1, 2 and 3.

The graph shows the poor prediction accuracy of all predictors when the relevance

parameter α refers to the same real world measure as the level of quantization. As expected at the same error range (for instance 10m, when $\alpha = 2$ or 1 for 5 and 10 metre spatial quantization respectively) the five metre quantization performs better for all predictors since all predictors were exposed to more information. While shown for completeness, accuracies above 20m can hardly be considered useful for most real-world applications using pedestrian movement, with even 10 and 15 metres doubtful for many applications. This effectively rules out the use of ten metre spatial quantization.

Under five metre spatial quantization the proportion of predictions within 5m is exceptionally poor, essentially being unusable. Considering predictions within 10 and 15 metres shows much more promise. When considering predictions within 10 metres the three predictors MM, NB-D and WN all perform poorly comparatively to the SEQ, NB, NB-N and NB-O predictors. The same observation holds considering predictions within 15 metres. The MM-V predictor's prediction accuracy falls somewhere in the middle although as the level at which a prediction is considered acceptable increases this method tends toward the prediction accuracy of the poorer performing set. This result is mostly expected with the MM being the standard Markov Model which does not approximate the full length history or encode direction, and the NB-D predictor being the Navie Bayes' predictor that does not encode direction. Not expected is the poor prediction accuracy of the predictor from [137] which generally performs worse than even the standard Markov Model. Since the approach is similar to the other baseline, the SEQ predictor, two possible reasons can be speculated. Firstly the WN predictor penalizes matches by a normalized distance score at each point, thereby reducing the relative effect of points being further away. Secondly it is possible that the combination of the support of each node with the distance function is detrimental, with a better approach being the simpler approach in carried out in SEQ where the distance function is used first and then the frequency information used as a second step. Further investigation, however, is needed and this is addressed in later work in chapter 5.

Considering the better performing predictors it is of note that the Naive Bayes' predictors perform almost as well as the significantly more computationally complex SEQ baseline, with both the NB and NB-N predictors outperforming it slightly once the admissible level of average error reaches and exceeds 15 metres. Considering the difference in runtime and storage complexities this highlights the potential of these methods.

Between the different variants of the relaxed naive Bayes' methods encoding negative

information shows no difference. Since encoding negative information effectively doubles the space requirements it is likely this should not be used. That said, its prediction accuracy on other data sets should be considered since other data sets may contain more cases which benefit from its use. Additionally the effect of resampling the input and historic data could be considered, since the formulation of the negative information assumes the trails are continuous in nature (see section 4.2.3). As previously noted, however, this may also introduce different types error. While potentially worthy of future investigation it is possible that the composite effects contribute very little to the overall rankings of potential predictions. Encoding order seems to provide a small increase in performance at a small increase in storage cost (see section 4.2.1) and is considered worthwhile. The biggest impact on prediction accuracy is the directional encoding, which as expected, improves performance dramatically. This is the result of encoding additional information and making it available to the predictor. The proposed use of the directional encoding coupled with the relaxed model therefore forms the core of the predictor developed in this chapter.

In figures 4.12 and 4.13 histograms of the individual prediction scores are shown. The graph shows the distribution more clearly, with no unexpected results. All predictors distributions have long tails and are skewed towards zero error, with the better performing methods more heavily skewed than the rest. Under 5 metre quantization the same general observations can be made as from the previous graph, with the predictors SEQ, NB, NB-N and NB-O clearly showing better prediction accuracy than the NB-D, MM-V and MM predictors. Under 10 metre quantization the histogram shows a slightly different story, with the SEQ predictor looking like it performs significantly worse than the NB, NB-N and NB-O predictors. However, this is an artifact of the visualization based on the histogram shown using bins of 0.5 units.

As previously noted the 10m quantization can not be considered of much practical use. Neither are the predictions from the predictors under 5m quantization when predictions are considered relevant within 5m average Fréchet distance. Therefore these cases are not examined further. For further examination predictions within 10 and 15m are considered under 5 metre spatial quantization ($\alpha = 2$ and $\alpha = 3$ respectively). Box plots of the sample level generalization error[9] as shown in each case in figures 4.14 ($\alpha = 2$) and 4.15 ($\alpha = 3$).

[9] the plot of the proportions for each training/test set pair

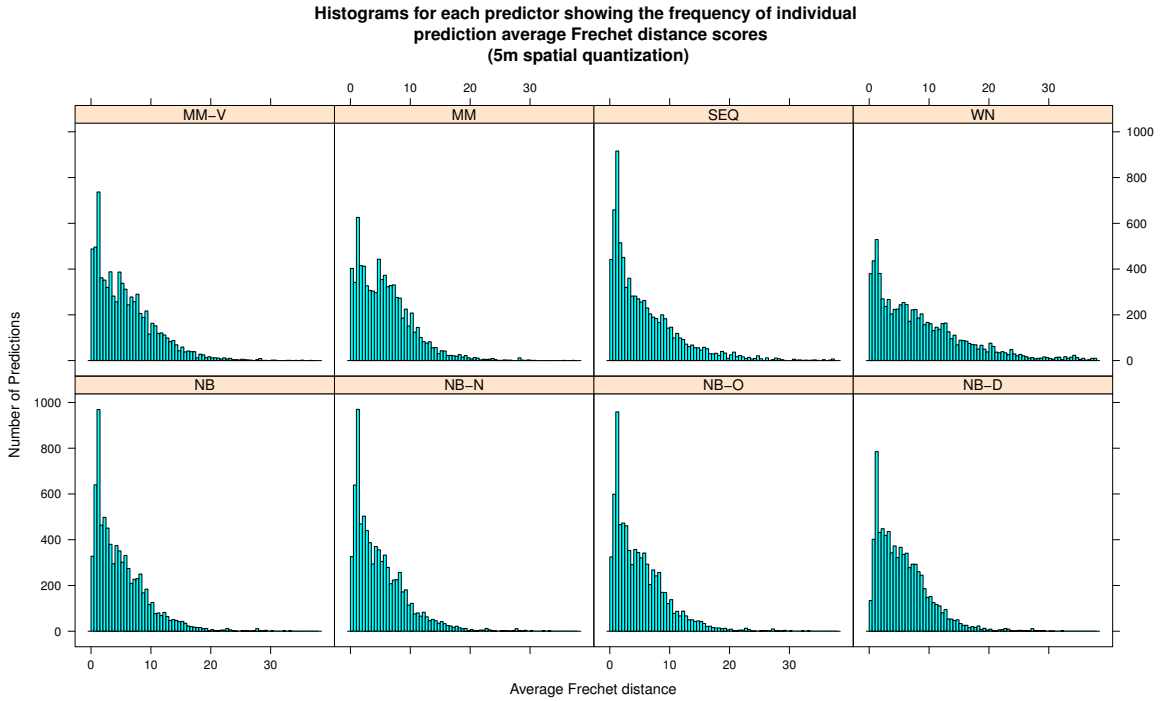


Figure 4.12: Graph showing side-by-side histograms of the individual predictions for each predictor under 5 metre spatial quantization.

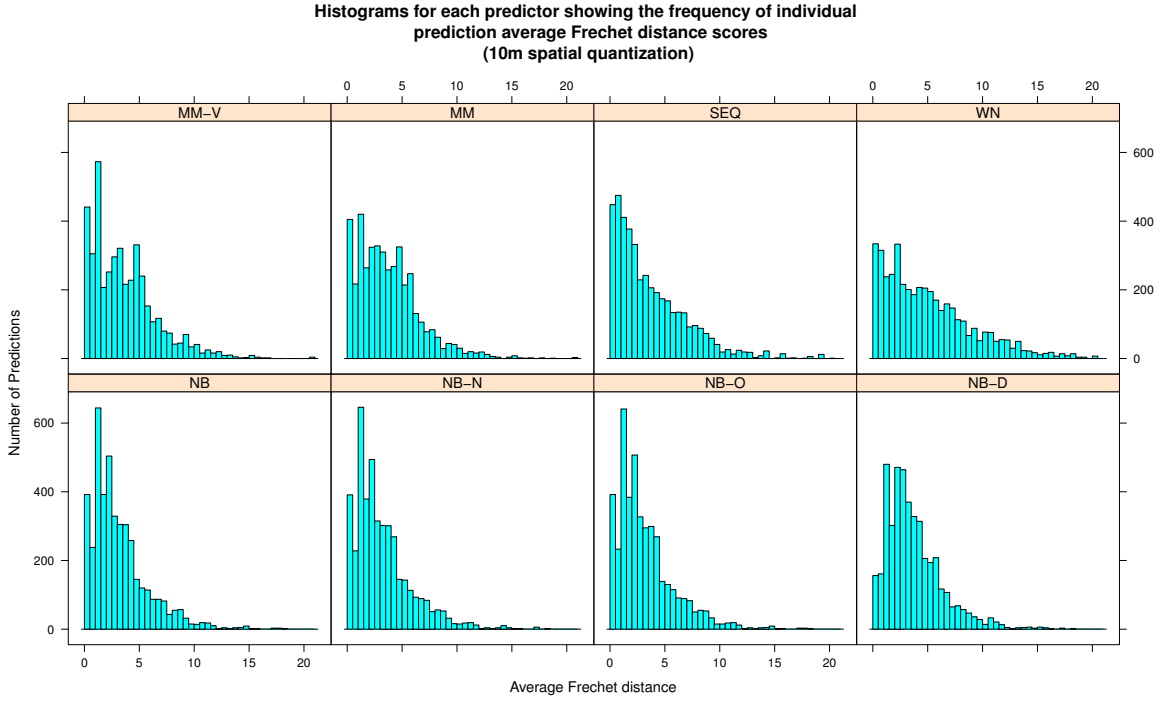


Figure 4.13: Graph showing side-by-side histograms of the individual predictions for each predictor under 10 metre spatial quantization.

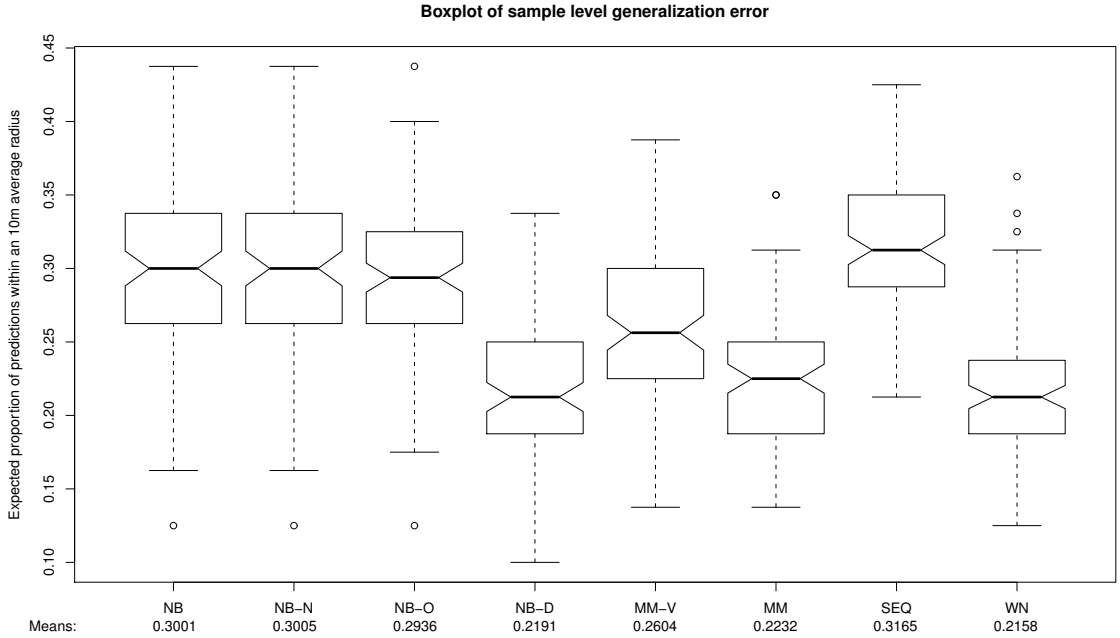


Figure 4.14: Side-by-side box plots showing the proportions from each fold in the cross-validation runs for each predictor under 5m spatial quantization with $\alpha = 2$.

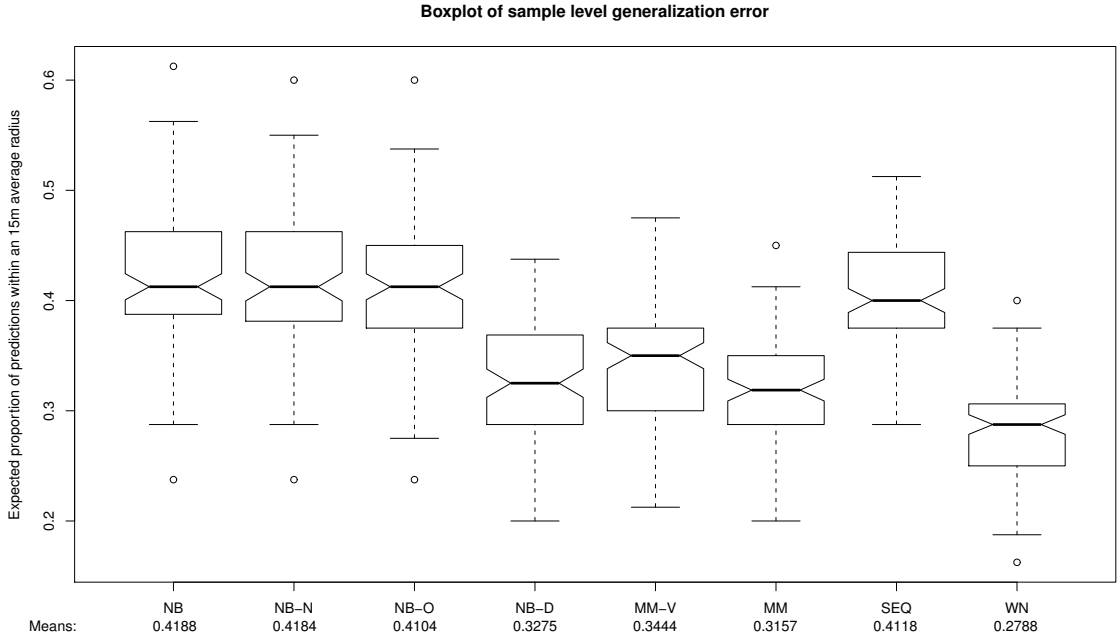


Figure 4.15: Side-by-side box plots showing the proportions from each fold in the cross-validation runs for each predictor under 5m spatial quantization with $\alpha = 3$.

From the box plots it seems unlikely that there is any significant difference between the NB, NB-N and SEQ predictors in either case which, considering the difference in complexity between the NB predictors and the SEQ predictor is an excellent result for the proposed algorithm. Additionally, it is almost obvious that there is a significant difference between these and the WN, NB-D, MM-V and MM predictors. The presence of a significant difference between NB-O and the SEQ predictors is less obvious. Statistical tests are performed based on the recommendations made in chapter 3. Tests were performed using paired t-tests in conjunction with a conservative estimation of the variance. Specifically the method OET detailed in section 3.2.3 was used to calculate the conservative estimate of the variance. The p-values were then adjusted via the Holm procedure. Before performing the statistical tests the distribution of each predictor was checked to ensure it satisfied the assumption of a normal distribution. This was done via a set of normal probability plots and the D'Agostino test of normality. These are shown in figure 4.16. In general the plots and the D'Agostino test support the assumption of normality. Only for the WN predictor when $\alpha = 2$ is the null hypothesis that this distribution is normal rejected by the D'Agostino test. However, the plot does not show excessive variation, and due to the underlying theoretical reasoning (central limit theorem) and the relative robustness of the paired t-test, the paired t-tests with the modified variance estimator are still used.

Tables 4.1 and 4.2 show all pairwise comparisons between the predictors and their adjusted p-values, via the Holm procedure, for $\alpha = 2$ and $\alpha = 3$ respectively. Before discussing the results it is important to highlight once more that the statistical procedure is conservative. In other words, just because a significant difference is not shown does not mean one does not exist. This is done to ensure that any significant differences reported are in fact significant (see chapter 3). It is also useful to note that the detection of a statistical difference is dependent on the difference between two predictor's mean accuracy over each identical prediction task and the variance of this mean as estimated by the conservative estimation procedure. Therefore certain pairs of predictors can lead to higher variance estimates and hence predictors which may seem to show a greater difference in the box plots may not report a statistical difference whereas others which look less likely to do. An example is the comparison between the NB-O and MM-V predictors and the NB-O and WN predictors. In the former a significant difference is shown but in the latter one is not. However, from the box plot it looks likely the opposite is more likely. On investigation the variance for the former is much smaller (approx.

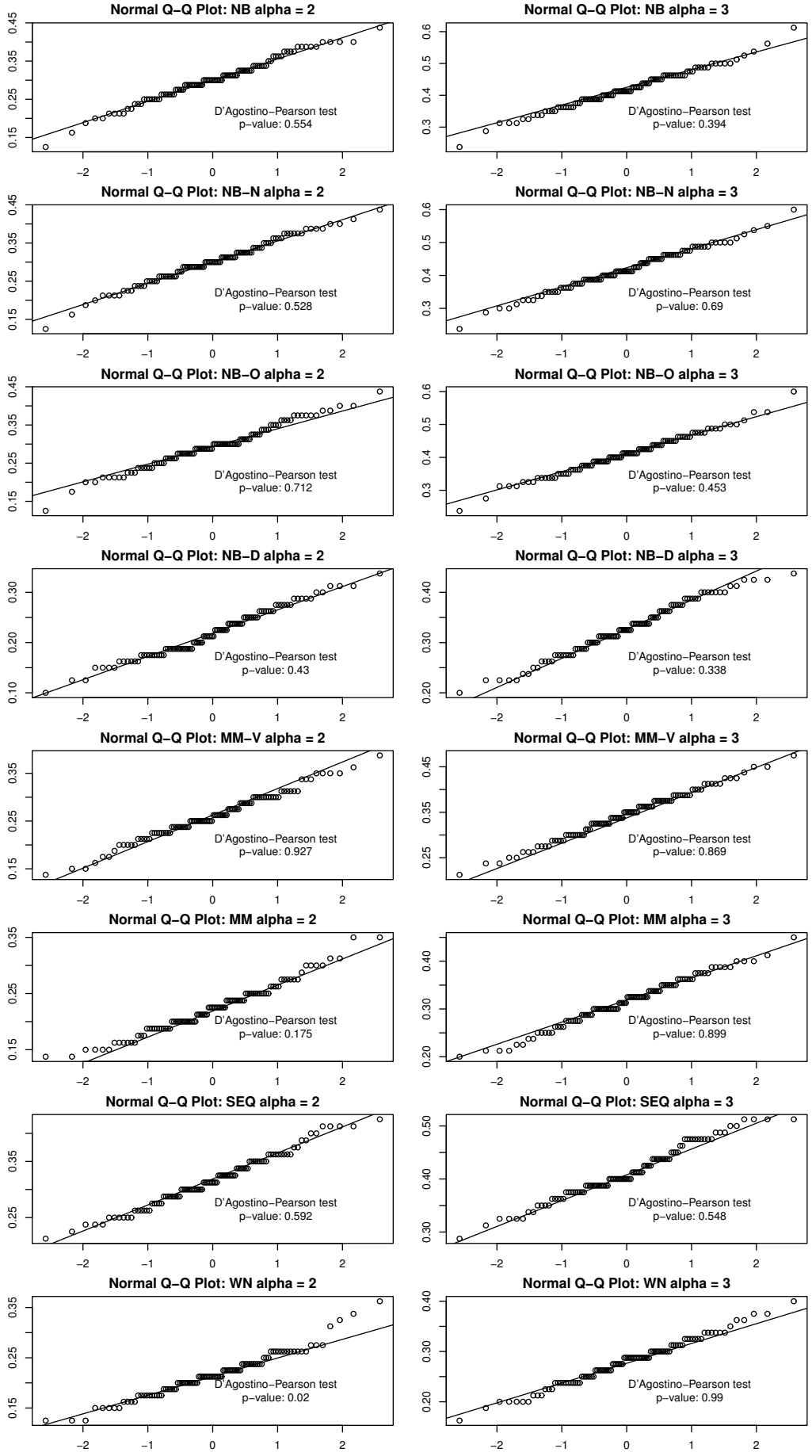


Figure 4.16: Normal probability plots and the results of the D'Agostino tests of normality.

0.0001 vs 0.0007). This once again highlights that only the significant results can been seen to carry meaning (that a difference exists) but non-significant results only highlight that one could not be shown.

Considering first the case where $\alpha = 2$, of the 28 comparisons, 17 show significant differences. Nine of these differences show what was to be expected from the box plot, that the predictors NB, SEQ and NB-N all significantly outperform the predictors WN, NB-D and MM ($p < 0.05$). Three show that the better preforming Naive Bayes' methods (NB, NB-O and NB-N) additionally all significantly outperform MM-V, the Markov Model predictor with direction information ($p < 0.05$). SEQ is not shown to outperform MM-V although it is close. A further two show that the NB-O predictor significantly out performs the poorer performing NB-D and MM predictors ($p < 0.001$). Of the final three differences two are shown between the Markov Model with direction over the two approaches without direction (MM and NB-D). This highlights the value of including directional information in the models. The final difference shown is between the Naive Bayes' predictor without negative information and the Naive Bayes' predictor without order. This indicates that encoding order is probably a good idea, although based on the box plot, the difference detected here is quite small. The rest of the comparisons show no statistical difference, although the pairs NB-O, WN and NB, NB-O come close.

When $\alpha = 3$ the separation between the better performing predictors (NB, NB-N, NB-O, SEQ) and the poorer performing predictors (MM, NB-D, WN and now including MM-V) increases slightly. Significant differences are shown between all in the former group compared to all in the latter. This highlights the reduced comparative predictive accuracy of the MM-V prediction which also no longer shows significant differences between the NB-D and MM predictors. A final point to note is that when $\alpha = 3$ the NB predictor slightly outperforms the NB-N predictor (although by no means significantly) and the significant difference shown between the NB-O predictor and the NB-N predictor is replaced with a significant difference between the NB and NB-O predictors.

Overall the results show that the newly proposed naive Bayes predictors (NB-O to a lesser extent) are significantly better than the traditional Markov approaches with similar computational costs. Additionally the new approach provided significantly superior prediction accuracy compared to the predictor from [137]. Statistically speaking the claim that there is no statistically significant difference between the naive Bayes predictors prediction accuracy and the more computationally expensive sequential predictor

Predictors			p-value	Adjusted p-value
better		worse		
NB	vs	NB-D	0	0
NB	vs	MM	0	0
NB-N	vs	NB-D	0	0
NB-N	vs	MM	0	0
NB-O	vs	NB-D	0	0
NB-O	vs	MM	0	0
SEQ	vs	WN	0	0
MM-V	vs	NB-D	0.000005	0.000102
MM-V	vs	MM	0.000007	0.000136
SEQ	vs	MM	0.000021	0.000393
NB-N	vs	MM-V	0.000086	0.001552
NB	vs	MM-V	0.000165	0.002803
SEQ	vs	NB-D	0.000838	0.01341
NB-N	vs	NB-O	0.001009	0.01514
NB-N	vs	WN	0.001339	0.018743
NB-O	vs	MM-V	0.001515	0.019696
NB	vs	WN	0.002224	0.026691
NB-O	vs	WN	0.005025	0.055279
NB	vs	NB-O	0.018781	0.187813
SEQ	vs	MM-V	0.02071	0.187813
MM-V	vs	WN	0.055025	0.4402
SEQ	vs	NB-O	0.409126	1
SEQ	vs	NB-N	0.539568	1
SEQ	vs	NB	0.545008	1
MM	vs	NB-D	0.699905	1
MM	vs	WN	0.734083	1
NB-N	vs	NB	0.859213	1
NB-D	vs	WN	0.902377	1

Table 4.1: Table showing the pairwise comparisons of the predictors (paired t-test using the conservative variance estimator from [141]) with both the p-value and the adjusted p-value (via Holm's procedure) when the relevance parameter, α, is set to 2 (10m). All p-values are shown to six decimal places. Better/worse column ordering is via means (see figure 4.14).

Predictors			p-value	Adjusted p-value
better		worse		
NB	vs	NB-D	0	0
NB	vs	MM-V	0	0
NB	vs	MM	0	0
NB-N	vs	NB-D	0	0
NB-N	vs	MM-V	0	0
NB-O	vs	NB-D	0	0
NB-O	vs	MM-V	0	0
SEQ	vs	WN	0	0
NB-O	vs	MM	0	0.000001
NB-N	vs	MM	0	0.000002
SEQ	vs	MM	0	0.000005
NB	vs	WN	0.000002	0.000038
NB-N	vs	WN	0.000003	0.000051
NB-O	vs	WN	0.000019	0.000282
NB	vs	NB-O	0.000082	0.001152
SEQ	vs	MM-V	0.0027	0.0351
SEQ	vs	NB-D	0.003509	0.042104
MM-V	vs	MM	0.006325	0.069579
MM-V	vs	WN	0.010367	0.103666
NB-N	vs	NB-O	0.033531	0.301782
NB-D	vs	WN	0.08604	0.688323
MM	vs	WN	0.093194	0.688323
MM-V	vs	NB-D	0.093762	0.688323
NB-D	vs	MM	0.451834	1
NB	vs	SEQ	0.794018	1
NB-N	vs	SEQ	0.810021	1
NB	vs	NB-N	0.871833	1
SEQ	vs	NB-O	0.960487	1

Table 4.2: Table showing the pairwise comparisons of the predictors (paired t-test using the conservative variance estimator from [141]) with both the p-value and the adjusted p-value (via Holm's procedure) when the relevance parameter, α, is set to 3 (15m). All p-values are shown to six decimal places. Better/worse column ordering is via means (see figure 4.14).

(SEQ) can not be made. Equally, however, the claim can not be made that the SEQ predictor is superior and considering the prediction accuracy on this data set as indicated by the box plots and relevance parameter graph it is clear the best Naive Bayes' predictors have similar levels of prediction accuracy, with the naive Bayes' even showing slightly higher mean accuracy when the acceptable error level was equal to or greater than 15 metres.

In summary the results show that the proposed method is the clear choice when prediction needs to be performed on systems with minimal resources, significantly outperforming other methods of similar complexity and rivaling approaches which requires significantly more resources.

4.5 Conclusion

The application of movement prediction within mobile intelligent agents calls in many cases for computationally efficient algorithms that can operate on platforms with limited resources. This chapter presented a novel predictor based on approximating a full order Markov predictor, while still using resources in the order of a simple first order Markov Model. A number of variants were also considered with the experimental results indicating that encoding direction and a notion of future and past provided improvements (statistically significant in the first case) but that encoding negative information was of limited use, at least in this data set.

Compared to predictors of similar complexity the novel predictor showed significant improvements in prediction accuracy. Significant improvements in predication accuracy was also shown with respect to the predictor proposed in [137] and the approach rivaled the prediction accuracy of an enhanced baseline which utilized both full order history and a distance function to find the best match to the input. Compared to this sequential predictor the proposed approach utilizes substantially less space, with requirements in the number of unique locations seen, with an upper bound in the square of the number of possible locations. This is in comparison to variable space with an upper bound on the number of unique possible sequences. Additionally the proposed approach does not minimize a global distance function resulting in a considerably reduced runtime, with the number of multiplications required in the order of eight times the length of the input rather than in the number of distance calculations between even a subset of all historic

paths[10].

In summary, the proposed model provides a state-of-the-art approach for pedestrian route prediction from GPS data under limited resources, a category which includes most mobile devices, significantly outperforming predictors of similar complexity and rivaling the prediction accuracy of much more complex models.

[10]The brute force approach requires distance calculations between each point in the input and each point in each historic trail, however, tree based data structures and heuristics can greatly reduce this. However, the use and impact of heuristics on prediction accuracy have not being examined in the literature. The brute force approach was utilized in the experiments within the chapter.

Chapter 5

Beyond conditional probability and ranked lists

This chapter is joint work with Dr. James Goulding from the Horizon Digital Economy Institute at the University of Nottingham. The work was undertaken while visiting the Horizon Institute under the Maurice de Rohan Scholarship.

In large data sets routes often appear multiple times. Intuitively the more times a route is seen the more likely it is to occur again, and therefore it is potentially more useful to use these more frequent routes as predictions where there is no additional information. The mining of frequent routes can be generalized to the mining of rules in the form of $A \rightarrow B$ where the prediction process then matches the input sequence to A and uses B as the prediction. The most common approach is to score such a rule using conditional probability. Within the movement prediction literature, only conditional probability and frequency have been considered despite over 30 different measures being proposed within the data mining community [71]. Examining this, this chapter provides an in-depth study motivating and selecting five different measures most relevant to movement predictions and considering 30 different variants with respect to the utilization of the measures in conjunction with selected matching methods within the prediction algorithms. Enabling this, a new predictor is proposed which relaxes the sequential constraints addressing the issues of noise (see section 2.2) and permits the direct application of a wide range of alternate objective functions in order to assess their performance with respect to the baseline of conditional probability. The chapter also forgoes the typical standard output of movement prediction algorithms, the ranked list

of predictions, instead placing a larger onus on the importance given to each potential trail by combining them into a "heat map" (more formally a *probability mass function*) as the evaluated output. As such the probabilistic distribution of the subject's future position, as indicated by the importance measure, is evaluated. Therefore the chapters contribution is three-fold, firstly contributing an investigation into different measures for identifying useful prediction rules from large data sets, secondly examining a new prediction approach based directly on data mining approaches. Finally a new form of output is motivated and an evaluation procedure for such output proposed.

5.1 Prediction as a form of association rule mining

Movement prediction via mining movement logs has seen a wide range of techniques applied. In this chapter predictors are considered within the *association rule mining* framework popular within the data mining community. Association Rule Mining attempts to model a dataset by reducing its important relationships to a set of material implications of the form $q = a \rightarrow c$, with a being referred to as the *antecedent itemset* of the rule and c its *consequent itemset*. The technique was originally introduced in [6] to ascertain relationships between consumer purchasing decisions, with rules being accompanied by an objective function to measure the applicability of any rule generated. Predictions are made by comparing an input set of items with the antecedents of all available rules, ranking matched rules by their objective values to produce an ordered set of predictions. The association rule conceptual framework clearly separates the matching of the input to the antecedent from the selection of the final rule based on a measure that considers the repeated observations (if any) of a route within the data. If anything, the latter is emphasised. This in contrast to previous discussion where the focus has been on the matching mechanisms. Throughout this chapter the selection measure will be referred to as the objective (value) function.

The primary goal of these approaches is to first mine the important patterns from the data to a candidate set, acknowledging that (1) the full data set may be so large that it is not always possible to consider all the information and (2) that not all information is relevant, or equally relevant to the task at hand, leading to the emphasis on the objective functions. Traditional data mining approaches therefore select and rank the mined rules

first and matching occurs as a separate second step, thereby reducing the instances for which matching needs to be performed, aiding in issues of tractability. Of course objective functions can still be used conceptually second, with matching occurring first, by ensuring all instances make it into the set of candidate patterns. Since the mining approaches are generally parameterized this is easily achieved.

Many of the algorithms discussed in chapter 2, particularly those detailed in the pattern matching section (section 2.1.3), follow the former approach. For instance [139] base their work on the *PrefixSpan* algorithm [152] which first mines sequential patterns from the data set. The predictor from [95] also aims to build rules using a modified version of the main association rule mining algorithms, the apriori algorithm [6], based on fixed length time aligned trajectories. More recently [137] propose a predictor based on temporarily annotated sequences mined using algorithms from [74–76]. Operating in a discrete data space, the approach is based on generic mining algorithms which aim to identify frequent (sub)sequences [74–76, 152] or frequent sets [6] from which rules can be constructed.

In general association rule mining are symbol based algorithms which have three main steps in common: (1) some form of spatial symbol quantization, (2) rule mining and objective value calculation and (3) rule matching, ranking and selection. While large variation exists in the matching methods and encoding schemes used (see chapter 2, mainly section 2.1.3), the same cannot be said for objective functions. In fact only two objective value measures have been investigated despite the fact that over 30 exist [71].

5.2 Objective functions beyond support

Despite the application of a wide range of techniques applied to movement prediction from the data mining domain almost all have focused on the use of *confidence* (conditional probability) as the objective value function (e.g. [31, 95, 139]). This is despite the fact that over 30 different objective value functions exist [71] aimed at addressing a number of shortcomings with confidence and support in many situations. It is of note that the Bayesian predictors and the Universal predictors all also inherently use confidence as the objective function.

Within the association rule mining literature objective functions are used to identify

which rules to keep or discard, and how retained rules are ranked. The value is often intended as a measure of the rule's "interest" to the end user, and is often application specific. In [71] nine criteria are defined for which proposed objective value functions met a varying subset, highlighting the broad scope of association rule mining. The nine criteria are *conciseness*, *coverage*, *reliability*, *peculiarity*, *diversity*, *novelty*, *surprisingness*, *utility*, and *actionability*. Clearly not all are desirable from the perspective of movement prediction. For instance peculiarity, which refers to the desire to identify unusual or outlying behaviours, is clearly not of interest in the movement prediction domain although it is of primary focus in security applications, where unusual behaviour is desired to be flagged for further inspection. Of the nine criteria, coverage, reliability, utility and actionability are those most desired in the context of movement prediction. Of these coverage represents the patterns that account for a large number of observations, reliability represents the rules measure of conditional probability and utility and actionability refers to whether the patterns contributes to the overall prediction decision and future movement decisions. It is of note that the last two are more vague desires corresponding to specific applications of the movement prediction system, and that no general measures exist in these cases. Therefore discussion will focus mainly on the first two points. In addition some level of diversity may be desired, ensuring a list of potential predictions that all provide sufficient additional information to be worth presenting to the user (or utilizing system). This may be particularly important if the results were being displayed as map overlays for a human user with respect to managing information overload.

Considering that the main properties identified from the list in [71] with respect to movement prediction were coverage and reliability, it may seem intuitive to look no further than conditional probability when predicting movement from log data. However, the decision is not that straightforward, with many authors highlighting potential weaknesses in the use of confidence as an objective measure [71, 124, 184]. A first objection stems from the fact that correlation is often of interest over and above pure conditional probability. For instance, consider two locations, X and Y, within a large set of movement behaviour. Table 5.1 details occupancy of these locations over a set of historic movement trails, with the columns x and $\neg x$ corresponding to trails that did or did not, respectively, enter location X and rows y and $\neg y$ corresponding to trails that did or did not enter location Y (this is referred to as a contingency table).

	x	$\neg x$	Total
y	144	42	186
$\neg y$	514	36	550
Total	658	78	736

Table 5.1: Example of potential misleading conditional probabilities

From Table 5.1 it can be seen that the conditional probability is:

$$Pr(x|y) = Pr(x, y)/P(y) = 144/186 = 0.77$$

The example seems to indicate a pretty strong relationship between locations X and Y - that is until one recognizes that over 90% of the trails observations also occupied X at some point in time (658 of the 736 recorded trails). This means that a person who has been observed to go through y actually has 13% *less chance* of going into X, than another person for which there exists *no information either way* concerning their interaction with Y. Not only is there a negative correlation at work here that has been disregarded by the objective function, there is consequently a potential source of error in the presence of noisy or incomplete information.

A second problem with conditional probability lies in the fact that it is a purely descriptive measure - its value remains unchanged when all the observation counts are multiplied by a constant $k > 1$ and this may not be desirable. Consider two rules $q_1 : a_1 \rightarrow c_1$ and $q_2 : a_2 \rightarrow c_2$ where $Pr(c_1 \mid a_1) = Pr(c_2 \mid a_2)$. These rules have equal strength, but in reality one may be more trustworthy than the other. Imagine, for example, that a data set contains one thousand instances of the trail represented by q_1, but only one instance of the trail represented by q_2. In this case it is intuitive to be more confident of predictions made using q_1 compared to q_2, since the former seems to indicate a robust account of reality whereas the latter is potentially a once off observation, possibly made due to noisy or faulty equipment. Unfortunately confidence does not reflect these concerns, providing equal objective values in each case. *Statistical objective functions* attempt to address this issue by actually considering sample sizes when rules are being weighted (often in an analogous fashion to statistical testing techniques), and these approaches represent a growing area of interest in rule mining. For example [113] state that "it seems logical to prefer statistical measures, as the reliability of its assessment increases with n, the number of transactions".

The above examples provide evidence that confidence can not simply be considered the

perfect choice for the objective function within the domain of movement prediction. To this end this work provides a set of first results in the investigation of a range of objective functions with respect to movement prediction accuracy. Extensive comparisons between the properties of different objective functions has been undertaken in [71, 184] and the evaluated objective functions are chosen in light of their analysis. As a first condition only functions with a co-domain between zero and one were selected in order to allow a common probabilistic interpretation of the output for comparative evaluation. Importantly all selected functions, apart from Support (which is included due to its importance in the literature), accord to the definition of normalized probabilistic quality measures (PQMs) [51] and hence are ideal for generating probability mass functions (heat maps) as output.

The selected objective functions for evaluation are:

1. **Confidence :** The conditional probability of a rule under consideration, and the most utilized objective measure in rule mining [6]. Given a rule $q = a \rightarrow c$, $Conf(q) = \frac{Pr(a,c)}{Pr(a)}$.

2. **Laplace :** The Laplace function is a variant of the Confidence function, and often preferred when determining the accuracy of an association rule [193]. It is defined as $Lapl(q) = \frac{Num(a,c)+1}{Num(a)+k}$. k is user-defined and set to 2 in this work.

3. **Cosine :** A widely-used similarity measure for vector-space models, monotonically related to Pearson's correlation [184]. In terms of other measures used in association rule mining it represents the geometric mean between Interest and Support. It is defined as $Cos(q) = \frac{Pr(a,c)}{\sqrt{Pr(a)Pr(c)}}$.

4. **RLD :** Relative Linkage Disequilibrium is a statistical measure, originally proposed to analyse two-way contingency tables [104]. It is an adaptation of *Lift* that accounts more effectively for the deviation of rule Support compared to that expected under independence. $RLD(q) = \frac{D}{D-Pr(\bar{a},\bar{c})}$ where $D = Pr(a,c)Pr(\neg a,\neg c) - Pr(a,\neg c)Pr(\neg a,c)$.

5. **Support :** Support is the joint probability of the rule's antecedent and consequent. Despite its simplicity it is often used in rule mining due to possessing a downward closure property that allows pruning of the exponential ruleset search space [6]. Formally, $Supp(q) = Pr(a,c)$.

Laplace has commonly been used to rank rules for classification [42, 193] and is chosen

as it provides a trade-off between reliability and coverage. Consider the formulation $Lapl(q) = \frac{Num(a,c)+1}{Num(a)+k}$. In the case of a small number of observed instances (low coverage) the result tends toward the parameterized probability $1/k$. In this way the Laplace value is sensitive to a constant multiplier across all observation counts. From a computational perspective the Laplace measure is monotonic in both support and confidence, a property that helps reduce the computational costs when only the highest ranked rule is required [71]. Within the domain of classification rules the parameter k has generally been set to the number of classes. In this work, k is set to 2 following definitions from [71, 184]. In contrast, the Cosine and RLD measures can be seen to incorporate a measure of the deviation from statistical independence (with respect to the occurrence of the antecedent and consequence). More specifically, the Cosine measure is the geometric mean between a measure of the deviation from statistical independence and support. RLD is a more recent proposal that aims to account more effectively for the deviation of rule Support compared to that expected under independence.

5.3 Holistic Route Prediction

In this section a new predictor based on mining and applying association rules, referred to as *Holistic Route Prediction* is detailed. The predictor forms the basis of the experimental investigation into the selection of objective functions and input matching strategies. The method addresses prediction into a medium to long term horizon and intrinsically incorporates route estimates. The technique is holistic in the sense it considers the possible interdependencies across all positions in a trail, rather than an ordered subset. This is important as it provides a principled way of performing predictions which are insensitive to incomplete data and computationally tractable with large data sets, while allowing variable-length histories to be represented, unlike traditional Markov Models. Additionally, and of importance to the experimental investigation of objective functions, it allows the direct application of all objective functions from the literature without modification. As previously mentioned the output from the predictor is not a list of possible next steps, but rather a "heat map" (more formally a *probability mass function*, or pmf) indicating the relative likelihood of future occupancies based on the given input. Such an output is used for two main reasons. The primary motivation is that it allows a principled way of evaluating more than just the first prediction. Specifically it allows all prediction rules matched, and the way each was weighted by the

objective function, to play a part in the overall score, rather than just the top prediction. The second point of motivation is more practical with respect to the evaluation of the objective value functions. Since the model relaxes the notion of time in order to address issues of noise, and to enable the direct application of objective value functions from the data mining community, a strategy to recover a route for prediction is required. As discussed previously in chapter 4 this can be done in a number of ways, with the most practical approaches being based on heuristics to ensure the fast runtime performance of the predictor. However, the choice of such a heuristic can have a significant impact on the result. Since the primary aim is the investigation of the impact of the objective value function, and coupled with the desire to evaluate more than just the first ranked prediction, it is desirable to forgo this task altogether. Finally it is of note that in some cases a pmf (heatmap) may be the desired output of a prediction system anyway, with the output easily visualized and intuitively understood by human operators.

As with most association rule mining techniques, the holistic route predictor mines and reasons about symbolic data. A number of quantization techniques were discussed in chapter 2 and in this case a simple grid-based quantization is applied. Therefore the spatial area of interest S is defined as:

$$S = \langle s_1, s_2, \ldots, s_{m \times n} \rangle \tag{5.1}$$

where each s_i is an individual, non-overlapping spatial region (i.e. a particular grid square), with each of these cells being indexed in an arbitrary, fixed sequence. Each movement trail, H, in the dataset, D, (represented as a set of temporally-indexed points, each encoded as a WGS84 long-lat coordinate, h_i), is then quantized against S:

$$\mathsf{h} = \left\{ x \in S : \exists h \in H, \forall s \in S, \left(\left\| \vec{xh} \right\| \leq \left\| \vec{gh} \right\| \right) \right\} \tag{5.2}$$

This process is illustrated in Figure 5.1. The aggregation of these quantized trails yields the final set of historical observations, D, which has a cardinality $k \leq m \times n$.

5.3.1 Constructing the Model

Implementation begins by iterating through the set of historical observations, D, and treating each trail, H as an *itemset* (I_{H}) of grid occupancy decisions. All possible rules are extracted from this itemset by combining its elements to produce material implications

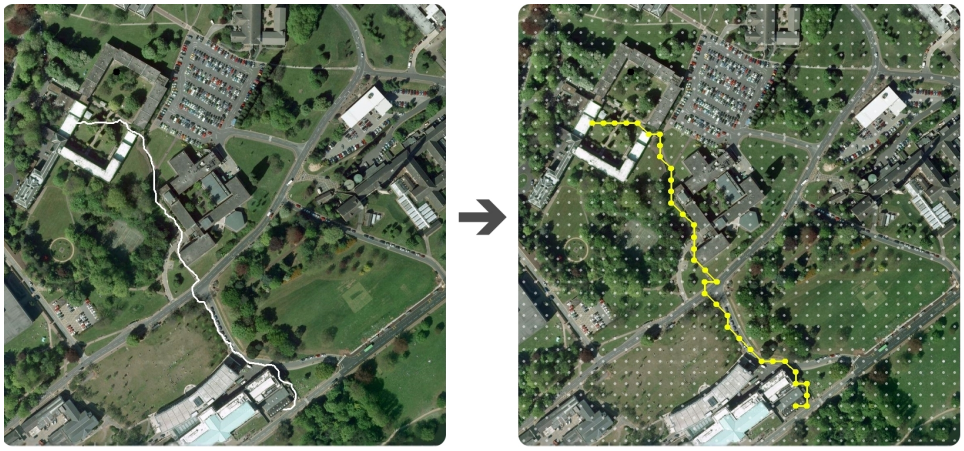


Figure 5.1: An illustration of the quantization process, with trail H being converted into quantized form (as described by equations 5.1 - 5.2), using a resolution of $10m$ and a 40×40 grid of points, S.

of the form $a \rightarrow c$, where a is a subset of I_{H}, c is a single element of I_{H}, and $a \notin c$. By iterating through each observation, and aggregating all of the rules generated, a set of rules for the whole data set is produced, as defined in equation 5.3:

$$Ruleset_D = \bigcup_{\mathsf{H} \in D} \{(a \rightarrow c) : a \subset I_{\mathsf{H}}, c \in I_{\mathsf{H}}, c \notin a\} \tag{5.3}$$

The rulesets that are generated by association rule mining can be very large. For any observation the number of rules generated is $2^n - 2$, where n is the size of the itemset, $\|I_{\mathsf{H}}\|$. As the lengths of observation trails increase, memory is consumed at an exponential growth rate of order $O(2^n)$. Because of this, restricted rulesets are often generated through use of the *Apriori* algorithm [6], which uses an iterative pruning technique to try and ameliorate these computational issues. Often this is done by harnessing the anti-monotonic property of minimum Support (the percentage of observations in the dataset which contain both the antecedents and consequents of a rule). The implemented approach employs this technique[1], but an extremely low Support threshold of 0.005 is used.

Finally, each rule, q, is assigned a weight via an objective function, $Obj(q)$. Traditionally this weight is known as the rule's "interestingness" but thoughout this thesis the term *objective value* be used. The objective value most commonly used in the literature is the conditional probability of a rule, $P(c \mid a)$, which is determined by extracting those observations from the dataset which contain the rule's antecedent and calculating the

[1]Specifically we use the implementation from the **R** package *arules* [85] which can be found at `http://r-forge.r-project.org/projects/arules/`.

percentage of these which also contain the antecedent[2]. However, conditional probability is just one of numerous different objective functions that can be utilized, as previously discussed.

The use of unordered itemsets has a useful side effect. Because rules are constructed from non-continuous trail sections objective functions will assign greater values to those that contain highly correlated squares than those containing infrequently observed squares. This results in a process of implicit noise reduction as rules containing outlying point observations are punished. If a threshold is also employed during rule generation so that only top-ranked rules are retained this noise elimination becomes explicit and the deleterious impact of small route deviations is drastically reduced. An illustration of this effect is shown in Figure 5.2.

As a consequence of using a non-sequential encoding, rules will often encode non-continuous movement patterns and because of this, input-matching strategies will have to accommodate the fact that exact matching of an input trail against a historical route is very unlikely. However, this should not be seen as a limitation, as variation in movements and noise inherent in GPS measurements already requires robust matching mechanisms.

5.3.2 Generating Predictions

Assuming the existence of a rule set predictions can be made by matching the input to the rule antecedents. Specifically Given a set, E, of observed region occupancies,

$$E = \{\mathsf{e}_1, \mathsf{e}_2, \ldots, \mathsf{e}_{|E|}\}, E \subset S \tag{5.4}$$

the aim is to construct a "heat map" (pmf) of future occupancies as the prediction output. Suppose that $\Pi : S \to \{0, 1\}$ is a discrete random variable, denoting the grid square occupied by the subject at some random point in the near future. Then the probability mass function, $f_\Pi : \{0, 1\} \to [0, 1]$, representing the weight of belief that $\Pi = \pi$ (i.e. that the subject will be in grid square π based on the input, E, being

[2]Note the similarity between Markov models and association rule mining when conditional probability is employed as an objective value. In this case it is possible to consider an association rule model as a variable-length Markov model where higher order states are encoded as sets of observations rather than ordered tuples. Of course the unordered nature of the itemsets used in this implementation, however, introduces vast differences between it and the Markov models.

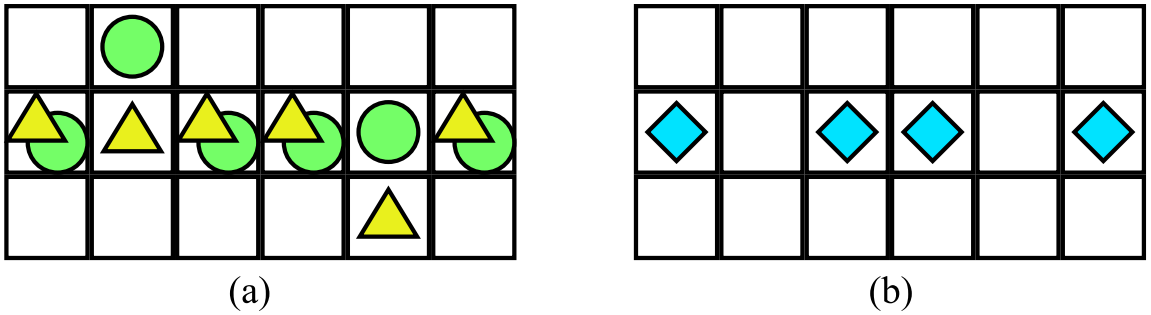


Figure 5.2: Two movement trails are shown in (a), represented by circles and triangles respectively. They are almost identical but differ slightly due to noise, a common occurrence in GPS observations. As a full trail each trail only has a *Support* value of 1, but the itemset made up of their intersection shown in (b) has a *Support* of 2. Any rules formed from the "noise-reduced" set in (b) will therefore be ranked more highly and have more impact on the resulting model.

projected into our ruleset, Q_D), is defined as:

$$f_{\Pi}(Q, E, \pi), \pi \in S \tag{5.5}$$

So long as an appropriate objective function is used in the generation of the ruleset, f_{Π} will correspond directly to a probabilistic distribution of the subject's future position (as is the case for all of the objective measures investigated in this chapter). The resulting pmf can then either be used programmatically or presented to the user as a heat map, an example of which is shown in Figure 5.3.

The first step in generating this output is to compare the input set, E, against the ruleset in order to find those rules that the input matches. In the simplest possible case there exists a single rule whose antecedent matches the input set of grid activations exactly. The resulting pmf would map the consequent to *one*, and all other grid squares to *zero*. If the input set matches the antecedent of more than one rule then the resulting pmf is only slightly less trivial, mapping the consequents of those rules to $\frac{1}{no.\,matched\,rules}$, and everything else to *zero*.

However, exact matching of this fashion is very unlikely. Even with $10m$ grid quantization, the noise from the GPS units means that even if a user follows the exact same route, the quantized trails generated are likely to be different - new input trails will not match historic ones exactly. This is referred to as the *Input Matching Problem*, and it is particularly prevalent in the fine granularity GPS data considered in this thesis. Strategies for dealing with this issue are discussed in section 5.3.3. The problem is formalized

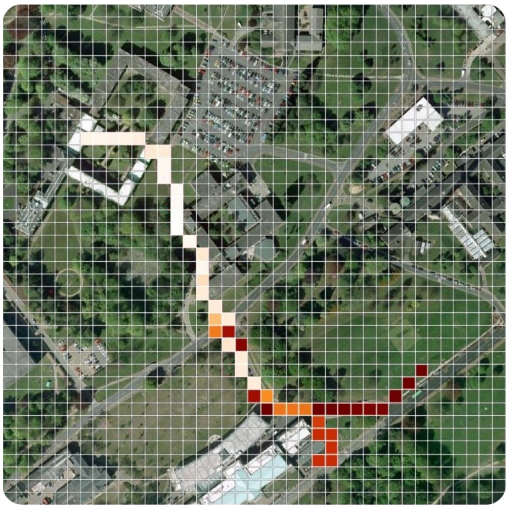


Figure 5.3: Graphical representation of the probability density function output of the model. Lighter squares represent higher probability of occupancy, darker ones lower probabilities.

by defining an arbitrary matching function, $Matches(Q, W) \subset Q$, that given the input, E, returns a set of matched rules when it is projected into the ruleset, Q.

Most previous approaches have only attempted to rank the results of this matching, but in this work it is desired to combine the matched rules to form a single prediction in the form of a probability mass function for evaluation. However, it is likely more than one matched rule will have the same consequent, making a statement about the same physical location at the same time. This situation is addressed in probabilistic fashion. Equal weights are assigned to all unique antecedents with the pmf mapping each grid square within the consequent to the mean of the matched rules' objective values. This strategy represents the notion that if there are multiple rules that match the input equally well, their consequent should contribute equally to the final prediction. In order to present this formally a function is first defined that, given a grid location, π, returns the subset of rules that have π as their consequent:

$$Cons(Q, \pi) = \{(a \rightarrow c) \in Q : c = \pi\} \tag{5.6}$$

This allows the definition of the *activation* of a grid square when a given input, E, is projected into the ruleset. This is calculated by taking all matched rules $M = Matches(Q, E)$, and finding the average objective value of the subset of these, $C = Cons(M, \pi)$, that have the grid square, π, in the consequent:

$$Act(Q, E, \pi) = \sum_{q \in C} \frac{Obj(q)}{\|C\|} \tag{5.7}$$

From these activations the probability mass function detailed in Equation 5.5 can be calculated. This is achieved by normalizing the activation function so that its outputs sum to one over the area of spatial interest, S:

$$f_{\Pi}(Q, E, \pi) = \frac{Act(Q, E, \pi)}{\sum_{s \in S} Act(Q, E, s)} \tag{5.8}$$

Because a non-sequential encoding scheme is utilized, the pmf heat maps returned as output offer no explicit chronological information. Although not the focus of this work, there are options for remedying this - for example, *post-hoc* processing can be used to determine timings to locations based on the observed velocity, cost surfaces and/or historic speeds though the area. Time could also be integrated into the framework proper, perhaps in an approach similar to [95] where quantized time symbols are exhaustively concatenated to the symbols representing the spatial quantization. The probability mass function returned would then suggest the relative likelihood that the space would be occupied at that time in the future based on the given input.

It is finally worth noting that any trail, such as E, can itself be treated as though it were a probability mass function, by applying the following transformation into functional form:

$$f_{\Pi}(E, \pi) \begin{cases} \frac{1}{|E|} & if\ \pi \in E \\ 0 & otherwise \end{cases} \tag{5.9}$$

This transformation is particularly useful when a prediction needs to be assessed, as it allows the direct comparison of the prediction pmf with the observed ground truth.

5.3.3 Matching functions as a parameter

In general the holistic predictor has followed the standard association rule mining paradigm. First a set of rules are mined and objective function values calculated. Recall a number of different objective functions have been proposed for evaluation. Typically prediction from association rules is made by simply matching (exactly) the input symbols against the antecedents in the rule set. As previously discussed (see chapter 2), this can not be considered an optimal procedure when dealing with low level GPS data, even when a relatively coarse level of quantization is used. In light of this a number of matching functions are proposed, with the holistic predictor effectively taking one of these as a parameter. The first four proposed can be considered *Spatial Matching Techniques* while the last two are *Hybrid Matching Techniques*.

Rule selection via Spatial Matching

The process of finding the best rule via matching the input (E) to rule antecedents is a conceptually simple task. In the naive case all rule's antecedents are exhaustively compared to the input and a distance function is used to calculated a score representing the distance between the two. The lowest scoring rule is then retained. If multiple rules have the same score these are all retained. As a secondary step the objective values from all the retained rules are used to generate the probability mass function (heatmap) as the output of the predictor. While the process is simple the choice of distance function is not, with trade-offs in accuracy and computational complexity in addition to varying theoretical motivations for different types of matching functions. As discussed in chapter 2 many different matching functions have been proposed. For completeness they are recapped here. The most straightforward approach is to only return a prediction if an exact match between the input and the antecedent can be found. While in [139] good results were obtained, it is generally considered that such an approach is flawed, leading to no predictions in many cases [69]. This is known within the universal prediction literature as the zero frequency problem [198]. Within universal predictors, matching has been performed by fallback[3] in the case of prediction by partial matching (PPM) [45] which was evaluated in [182] in the case of movement prediction from wireless models. A predictor using a maximum history length of two with a single level of fallback is recommended, although this is perhaps indicative of the domain where the very coarse-grained level of movement data means that the movement patterns of interest are fully expressed in such a short history. This is not the case in pedestrian route prediction where much longer, fine-grained, histories are common. A similar approach is realized in probabilistic suffix trees where the reverse matching is used. This process starts from the most recent point observation and continues matching until no match exists, effectively achieving prediction via fallback. In contrast, distance functions can be employed as in [137] where the L_2 distance function is used in conjunction with sequence to evaluate the distance between the input and the historic trail.

In the work this was achieved by computing the pairwise difference between points in the trails according to sequence indices. Similarly [69] use a custom distance function based on the Hausdorff distance to calculate the distance between the two trails. Again order is

[3]Recall that fallback refers to the repeated reduction of the input history by the least recently observed symbol in the case that a prediction can not be made from the full history.

utilized. In this work, four spatial matchers are evaluated. The first is the exact matcher which is utilized as a baseline. The second is similar to PPM using fallback (fixed escape probability of 1). Since only relative differences in predictions are of concern escape probabilities do not affect the outcome. The third uses the Mallows distance metric. This is selected due to the holistic view taken in this chapter. Specifically no order information is assumed meaning that the distance functions from [69, 137] can not be applied. Often used in image matching under a variant called the Earth Movers Distance (EMD) [166], the metric can be thought of as the unit average of the minimum amount of work[4] required to move the input's mass to form the antecedent (or vice-versa if the input is the shorter of the two). Leftover mass is traditionally either simply discarded, or penalized by a fixed value [153]. However this requires the selection of an arbitrary penalty value, which can seriously bias prediction evaluation. To avoid this we convert both input and rule antecedents into pmfs (as described by Equation 5.9). By definition this means that activations are normalized, and in such a situation the EMD degenerates into the Mallows distance [117][5]. This is especially important since GPS noise can cause two identical physical journeys to produce inputs of different lengths. The final matcher seeks to relax both the fallback mechanism and the distance metric based approach. The approach ranks the rules by the amount of intersection the antecedent has with the input, normalized by the maximum of the input or antecedent length. This allows antecedents to be matched despite gaps or unseen symbols appearing in the input. Therefore the approach provides a distance measure based purely on the symbol intersect operations ensuring cheap computation. All four matching methods evaluated in this chapter are listed below:

1. **EXACT matcher:** The simplest strategy. Implemented as a baseline, this strategy will only return a result if the input exactly matches a rule antecedent. Grid squares corresponding to the consequent of matched rules are assigned a value given by the objective function. From this a pmf is formed.

2. **FALLBACK matcher:** Attempts to exactly match the whole input. If this is not possible, the least recent symbol is dropped and an exact matching attempt is made again. This process is repeated until either a prediction is made or no symbols are left. If no symbols are left no prediction is made.

[4] distance $\times$ mass

[5] In our implementation we use the equivalent [117, 153] Earth Movers Distance implementation from `http://www.cs.huji.ac.il/~ofirpele/FastEMD/code/`

3. **MALLOWS matcher:** Compares normalized versions of the input against rule antecedents using the Mallows Distance.

4. **INTERSECT matcher:** Ranks the rules by the amount of intersection the antecedent has with the input. The highest intersection ratio is determined, with any rules attaining that value being retained.

The four matching methods provide a wide range of matching techniques over which to compare the impact of the objective value functions detailed earlier in this chapter.

Rule selection via Hybrid Techniques

Rule section by spatial matching reflects a process where the matching and objective values are used separately, with the spatial matching taking precedent. In contrast it is possible to consider both the spatial matching score and the objective value at the same time using a weighting function to arrive at a final ranking of predictions. The hybrid predictor then uses the hybrid scores when merging the top predictions to form the probability mass function (heatmap), although in general the combination of the spatial matching and objective function means that multiple rules are less likely to have the same score and in many cases only one rule is used to form the heatmap. In this chapter the two best performing spatial matching strategies are converted to hybrid techniques and their performance evaluated under all objective functions being investigated. The hybrid strategies evaluated are detailed below:

1. **HYBRID-INTERSECT matcher** The hybrid intersect matcher weights the objective value proportionally to the percentage of the intersection between the antecedent and the input. Specifically, given the input E, a rule with antecedent a has its objective value, $Obj(q)$, weighted by the ratio $\frac{E \cup a}{\max(|E|,|a|)}$. This technique is similar to the work presented in [95], except they additionally penalize time separations. If the input does not intersect with any antecedent then no accurate prediction can be made, in which case a pmf of equal occupancy probabilities, $\frac{1}{\|S\|}$ where S is the set of grid squares, is returned.

2. **HYBRID-MALLOWS matcher:** This method alters the objective value based on the Mallows distance function. Unlike the intersection-based approach, the Mallows distance function returns a value from zero through to infinity. This raises the question of exactly how much to weight the objective value per unit

increase in the distance between the input and the antecedent. Given an input, E, and an antecedent, a, separated by a Mallows Distance, MD, it has been chosen to weight the objective value by $\frac{1}{MD}$ unless they match exactly, in which case the original objective value is retained.

Compared to the hybrid intersect matchers, this proportional approach results in the harsher penalization of the small spatial deviations relative to large spatial deviations - e.g. a separation of two incurs a 50% penalty to the objective value, but further doubling this distance then only incurs an additional 25% penalty. Matches that are further away but with higher objective values are hence preferred when exact matches are not possible. It is of note that this method shares some similarities to that presented in [137] which, in conjunction with a temporal-based penalty where applicable, penalizes the Support of a prediction. However, the HYBRID-MALLOWS approach allows the use of the whole range of objective value functions considered in this paper.

5.4 Experimental Results

Evaluation consists of two parts: each objective function detailed in section 5.2 is examined in light of the spatial matching strategies described in section 5.3.3 and second, the hybrid matching strategies also described in section 5.3.3 are examined, again under all objective functions. In total the five objective functions and six prediction models result in 30 different variants. These variants are individually denoted as an ordered pair, $\langle Match, Obj \rangle$.

In the evaluation the D-SCENT data set was used, which features 12 locations over a $80,000m^2$ spatial area and involving trails from 60 participants. GPS data from each participant was collated every 5 seconds if they had moved at least 5 metres. Ten by ten meter grid quantization was used and the logs segmented into trails using a set of known destination locations. A total of 1570 movement trails were extracted, with an average a trail consisting of 8.655 spatial regions with a standard deviation of 3.39. Ten by ten meter grid quantization was used as a trade off between a realistic level of quantization for pedestrian route prediction and to ensure trail coverage and overlap similar to what would generally be expected in reality. This is important to enable the differences in the objective functions to be seen if they exist. Known-location segmentation was performed

in order to eliminate additional noise being introduced into the data set due to the use of a heuristic procedure, again to focus the evaluation on the impact of the objective functions.

For the experiments, the set of trails were encoded as described in section 5.3 (with all user information being discarded) and used to test the prediction models via ten by ten-fold cross validation. For each prediction a historic instance from the test set was selected and split at a random point into two parts. The first was used as the input (E) to the prediction algorithm and the second part used as the prediction ground truth (F). Each part was checked to ensure that it had at least one data point and was adjusted if it did not. Overall this resulted in 100 different training sets (each containing 1343 trails) with accompanying test sets formed from the held out data (157 trails). Therefore each of the 30 model variants were associated with 100 datapoints, where a datapoint represents the average performance attained by an individual fold (i.e. training set/test set combination).

Each individual prediction assessment consists of selecting a movement trail from the test set and randomly truncating it (to a minimum length of one) to form an input trail as previously discussed. Each model was then supplied with the exact same truncated input trail (E), and outputs a prediction. Since the prediction is in the form of a probability mass function (heatmap) the full evaluation procedure outlined in chapter 3 does not apply. Instead the prediction was assessed by computing the Mallows Distance[6] between the predicted pmf of future occupancy and the pmf constructed from the remainder of the test trail (F, as detailed in Equation 5.9), using the Euclidean distance as the ground distance. The Mallows distance [129] is used as it is a well known distance measure for comparing two probability distributions [79, 117]. If a method did not make a prediction an equal probability of occupancy was given to each quantized location representing a prediction where all locations are equally likely. Statistical evaluation of the differences between two predictors was performed by calculating an overestimation of the variance by the OET method discussed in chapter 3 setting $M = 7$ and using paired t-tests used in a pairwise fashion. The $p - values$ from the paired t-tests were then adjusted using the Holm procedure to account for the fact that multiple comparisons are being made. As motivated in chapter 4, the Holm procedure is used due to its availability within the **R** Project Statistical Computing [158].

[6]In our experiments we use the equivalent [117] Earth Movers implementation described in [154], available from `http://www.cs.huji.ac.il/~ofirpele/FastEMD/code/`

The results of the sample level generalization error is presented in the form of side-by-side box plots. Each point in the plot is the result of taking the average performance of one fold from the 10×10-fold cross validation. One box within the plot is presented per matching technique and objective function combination. Two plots are shown. Figure 5.4 shows the results of the spatial matching predictors. Figure 5.5 shows the results of the hybrid matching predictors.

5.4.1 Rule selection via Spatial Matching

Figure 5.4 shows the results of the spatial matching predictors. These are the predictors in which the spatial matching strategy was used to rank the rules based on how well the antecedent matched the input. The top ranked rule(s) were then used to construct the probability mass function of future occupancy (heatmap) based on the rule's score as provided by the objective function. With respect to the matching strategies, the results are in line with general intuition. The Exact matching strategy performed significantly worse than all others due to its inability to make predictions in many cases. The fallback predictor performed significantly worse than the Intersect and Mallows matching strategies. This result is as expected considering since both the Intersect and Mallows matching strategies utilize distance functions to select rule antecedents. Such an approach prevents noise in the input or historic observation causing the exact match to fail and the method to fallback to a shorter history. The use of a distance function also allows similar trails to the input to provide a prediction, even when there is no noise in the system. With respect to the Intersect and Mallows matching strategy, the more complex Mallows matcher performs significantly better ($p < 0.001$).

While differences exist between the different matching strategies, it is clear that the objective functions do not play a very significant role, with almost no differences seen between different objective functions within variants of the predictors with the same matching strategy. An exception is when using the Mallows matching strategy where the cosine measure shows a statistically significant difference in performance compared to confidence ($p < 0.01$), although the actual difference in mean performance is less than half a metre. A logical explanation for such a lack of effect in general could be that, when spatial matching is used first, in most cases only one rule is returned making the subsequent weighting by the objective function redundant. To investigate this two folds (156 predictions) were randomly selected and manually examined with respect to the

number of rules matched in the case of the Mallows matcher. The overall mean was 8.859 with a median of 6 and a standard deviation of 9.944. The distribution was positively skewed towards 1, with 18.5% of cases only returning one rule from the matching phase. The maximum number of rules present after the matching phase in the sample was 68. The fact that only one rule is left 18.5% of the time provides some explanation for the closeness of the results, however, since in over 80% of the time this is not the case it can be concluded that, at least in this data set, the objective functions assign similar relative magnitudes (since once the score are assigned they are then normalized in forming the pmf) to the trails discovered in the matching phase.

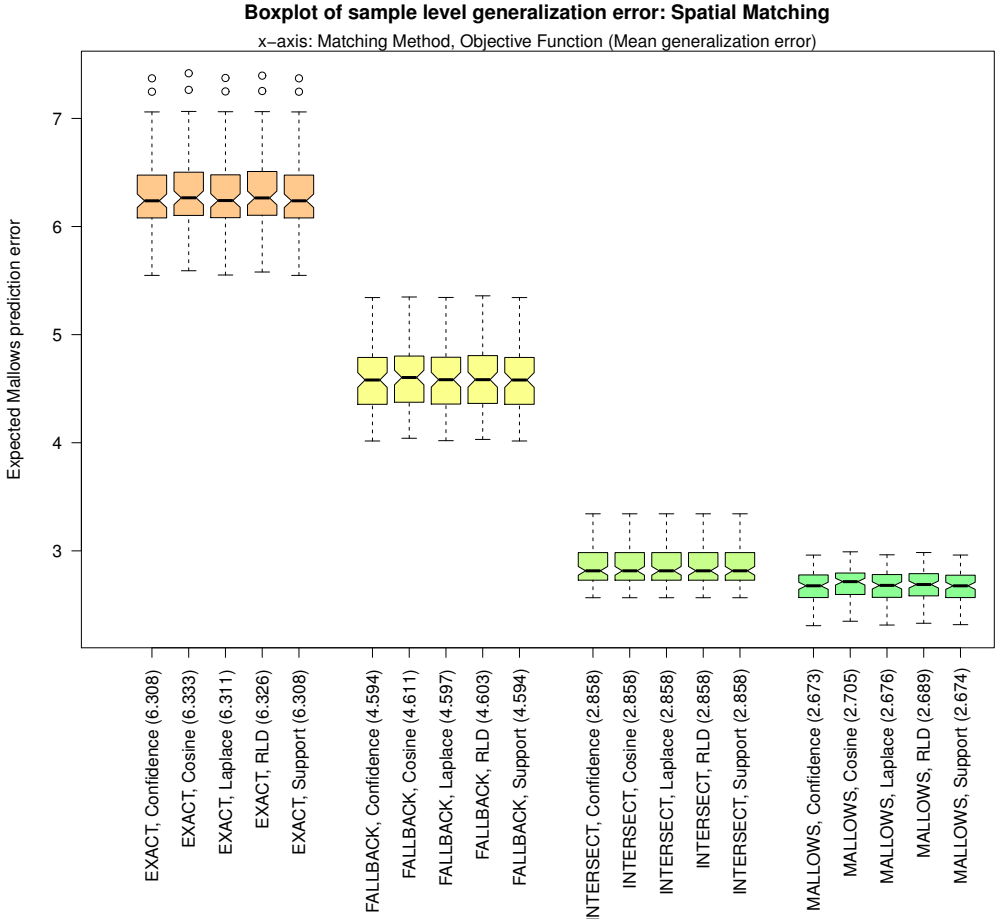


Figure 5.4: Box plot of the sample level generalization error for the predictors using spatial matching only for rule selection.

5.4.2 Rule selection via Hybrid Matching

Figure 5.5 shows the results of the hybrid predictors, including the corresponding spatial predictors for comparison. The hybrid predictors incorporate the rule objective value into the rule selection mechanism to create the heatmaps. This is done by multiplying

the two scores (matching and objective value) together after normalizing the matching function to range between zero and one, with one being an exact match. This is can be thought of as a re-weighting of the objective value for the rule based on how close the input was to the rule.

Compared to the rule selection via spatial matching only, it is clear that the choice of the objective value function does make a difference. This is due to the objective function being part of the score that selects the rules that form the pmf, rather than just the weighting of the selected rule predictions that form the pmf. Therefore small differences between objective value functions can cause a vastly different set of rules to be selected, and hence vastly different results.

Of the hybrid approaches the Mallows-based predictors show reduced prediction accuracy compared to their Intersect-based counterpart for the same objective value functions ($p < 0.001$, all cases). Within both the Intersect and Mallows variants the best performing objective value functions are Confidence and Laplace, with no significant differences shown between either under each matching function. When using a Mallows matching function Laplace shows a slightly better prediction average than Confidence and vice versa under the Intersect matching function, although the differences are very small.

Within the Hybrid-Intersect predictor, Confidence shows a statically significant difference in prediction accuracy compared to all others ($p < 0.01$) except Laplace and RLD. However, Laplace and RLD fail to show a significant difference between others such as Cosine ($p > 0.05$), only statistically showing a difference between Support ($p < 0.05$). Within the Hybrid-Mallows predictor, statistically significant results can only be seen between Support and all others ($p < 0.001$), with Support showing a much lower prediction accuracy.

5.5 Discussion

An interesting result of the experiments is the dominance of predictors that first select rules by spatial matching techniques ($p < 0.001$ in all cases). Approaches that integrate objective values into the matching process do not appear to improve route prediction, with the experimental results indicating that the opposite is in fact true. This result infers that matching and prediction processes are best treated as completely separate

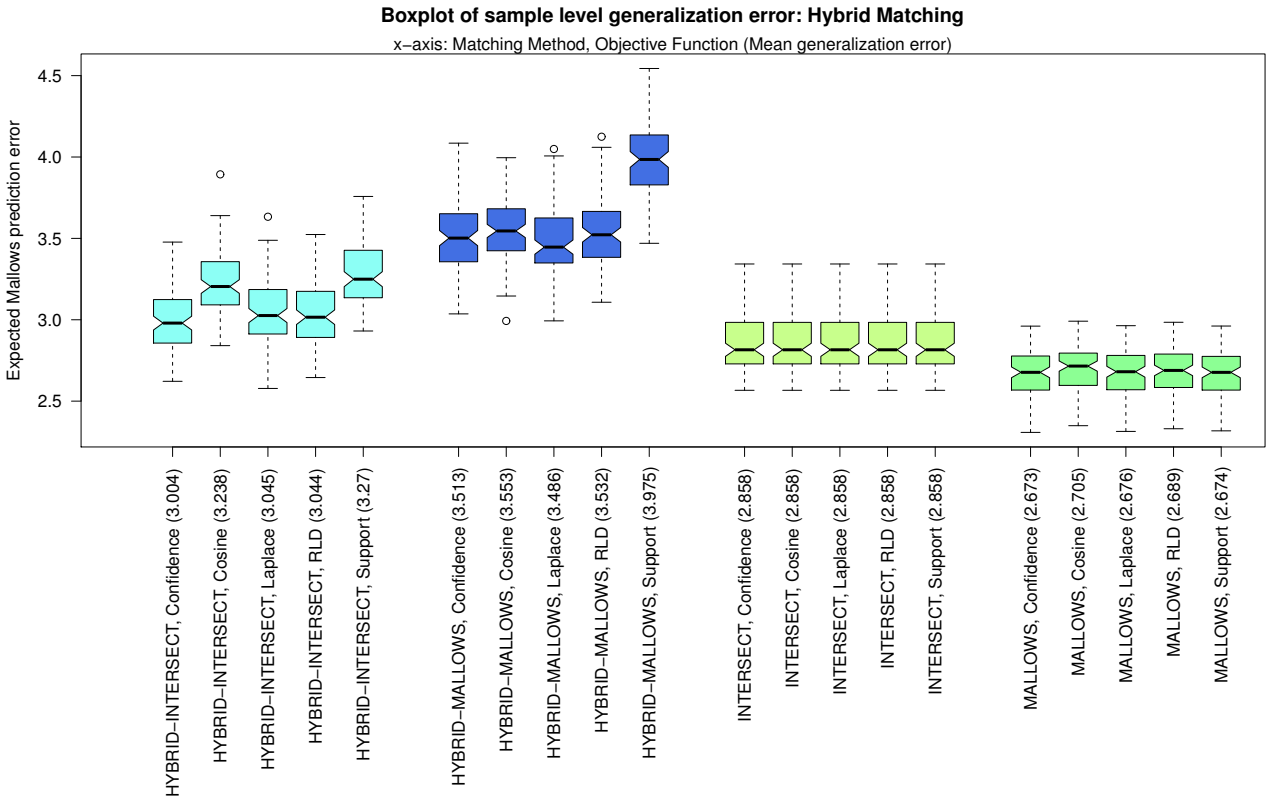


Figure 5.5: Box plot of the sample level generalization error for the Hybrid predictors. The corresponding spatial predictors are shown for comparison.

tasks. Moreover it demonstrates that rules close to an input should be prioritised over those that are further way, even when the distant rules are more highly rated. Note it is possible to parameterize hybrid approaches to a far greater extent than what was within the scope of this evaluation, so these interpretations must be approached tentatively. However, it is hard to justify the additional investigation due to the poor performance under the simple configurations evaluated and the complexity of choosing parameters that might provide further improvements. Nonetheless, rather than declaring the outright superiority of non-hybrid strategies, their good performance coupled with their ease of implementation and configuration is highlighted, additionally noting their robustness under the choices of objective value functions evaluated in this work.

In many ways this lends evidence against the base prediction algorithm utilized in this chapter. The proposed predictor was based directly on the apriori algorithm [6] from the data mining community. The algorithm allows sequence within the routes to be relaxed, building rules from the routes represented as sets by considering interdependencies across all elements in the sets. This approach addresses the issues of missing symbols, however, the resulting set of possible rules is much higher than when correct sequential information

is assumed, making the overall prediction process much more computationally complex in the naive case. The apriori algorithm deals with this by efficiently mining only the most important rules as defined by the objective value function and a user-defined threshold. This, however, is potentially equivalent to favouring rules that are rated more highly by the objective value function. Potentially this is at the expense of ones that may be closer but not as highly rated and fall below the user-defined threshold and never considered. While in the experiments the threshold was set to an extremely low value (0.005), in practice this may not be possible due to computational resource constraints and very large data sets. Therefore other algorithms that relax time, such as those proposed in chapters 4 and 6 are likely to provide better performance for the same amount of computational resources. Additionally this observation is applicable more widely. As discussed at the start of this chapter (see section 5.1) data mining techniques in general follow the approach of utilizing a user-defined threshold to reduce complexity and therefore this observation is an important one over a large range of predictors, including some that inherently model sequence (e.g. [95, 152]).

It is important to note that while not making a large difference in performance on the D-SCENT data set, the use of an objective function is still extremely valuable. Figure 5.6 shows a comparison of the Intersect predictor without using any objective function (in other words assigning a fixed score of one) in comparison to when the objective value functions evaluated in this chapter are used.

Considering no one objective value function stood out, two conclusions can be drawn. The first is that the choice of the objective value function makes no difference, and that while large theoretical differences exist between the function, so long as they all accord to the requirements of a probabilistic normalized quality measure[7], they have minimal practical consequence. This view is supported by the consistent performance of the non-hybrid predictors across the different objective value functions. However, it is somewhat refuted in the differences shown in the hybrid matching where significant differences are shown. The second possibility is that these results are tied to the characteristics of the dataset. Consider for example, the two better performing objective functions, RLD and Confidence. Although they performed equally well (particularly in the predictors based primarily on spatial matching), RLD is designed to look for statistical differences that may not have existed in our test dataset with its relatively even distribution of trails. If instead some trails were extremely well trodden while others were rarely recorded,

[7]As previously discussed, the class of objective value functions considered in this work

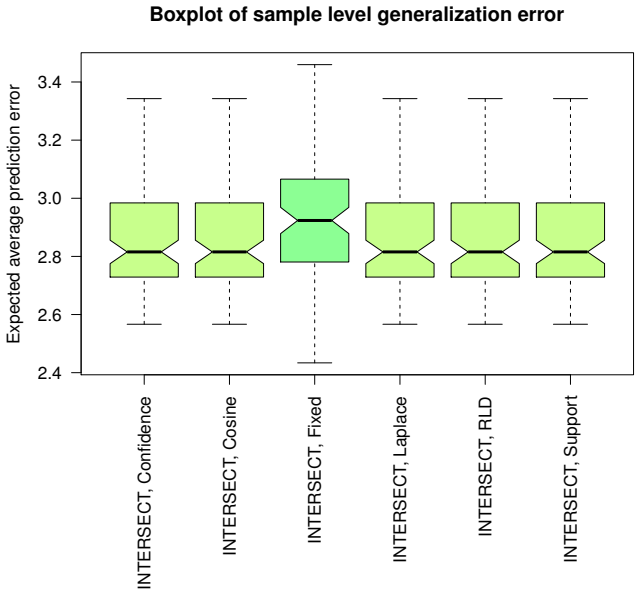


Figure 5.6: Box plot of the sample level generalization error for the predictors using spatial matching only for rule selection, including the case when no objective value function is used (INTERSECT, Fixed) and all rules are given a weight of 1.

RLDs scoring may have been more favourable. Such a hypothesis is only testable by contrasting results over distinct domains, and hence future experiments to contrast the effectiveness of models over different data sets are planned.

Of the two conclusions, neither is completely supported or refuted by the results. In general the results provide evidence that, at least when spatial matching strategies are used to select the rules initially, concern over the exact objective function need not be a primary focus and that the commonly used conditional probability is a good choice. As noted, however, it is not the only choice. Under certain data sets other objective value functions may perform better while performing no worse on data sets with similar properties to the data set evaluated in this study.

5.6 Conclusion

This chapter provided a comprehensive investigation into the use of objective value functions (measures) other than the traditional conditional probability for ranking potential predictions when multiple instances of historic instances (or part of the historic instances) matched the input equally well. In addition a probability mass function (heat

map) indicating the relative probability of future occupancy for each cell was considered as output from the prediction algorithms. Such an approach enabled the evaluation of more than just the top-ranked prediction and additionally enabled the evaluation to focus on the impact of the objective value functions. Based on theory from the data mining community a new predictor based on the apriori algorithm was utilized allowing five different object value functions to be directly applied. The approach also addressed the issues of noise and the computational complexity of taking a relaxed view on sequence, although the results of the study suggest the way this is achieved may be suboptimal. The framework also allowed the definition of spatial matching techniques as a parameter. This allowed the performance of the different measures to be evaluated under different spatial matching techniques. Finally two different ways of combining both the score derived from the spatial matching technique and the objective value function was investigated. Overall 30 different variants of the prediction algorithm were evaluated.

Experimental results showed that rules close to an input should be prioritised over those that are further away, even when the distant rules are more highly rated with respect to their objective value. This result highlights the need to ensure that data mining based prediction algorithms do not mine only the most frequent patterns and fail to consider less frequent patterns that could lead to more accurate predictions. With respect to the objective value functions under investigation it was found that when the previous advice was followed and close rules were heavily prioritised (with the objective values simply providing the relative probabilities in the output, the probability mass function) the choice between the objective value functions is of limited concern with all performing almost identically. However, as discussed such functions are still required. While the results showed that the commonly used conditional probability is a good choice, it must be noted that certain data sets with specific properties may benefit from other objective functions. An example is RLD which seek to address cases where conditional probability is known to give non-intuitive results.

Chapter 6

Modelling order: Augmented Cover Trees

In chapter 5 two of the main aspects of movement prediction via pattern matching were considered. The first was the use of various objective value functions. These provided a measure of how likely each prediction was based on repeated occurrences within the historical data set (measured by an objective value function, with frequency being the most basic). The second was the consideration of a number of simple matching strategies. The results showed that prediction rules spatially close to the input should be prioritised over those that are further away, even if the prediction rule has been considered more likely based on the score from an objective value function.

The results provide strong motivation for computationally efficient nearest neighbour matching algorithms which can take into account large numbers of prediction rules and as a result do not have to address computational complexity by mining a small number of frequent patterns determined by an objective value function. Based on this result this chapter proposes a computationally efficient approach that matches frequent patterns, although unless the number of historic observations is exceptionally large it is expected the algorithm will work off the historic observations directly. A major contribution is the extension of state-of-the-art by leveraging the observation that the assumption that accurate sequence information is present in the historic and input observation is not always correct. Using (1) an arbitrary feature encoding with respect to spatial, temporal and other aspects, (2) a pointwise distance (matching) function and, (3) global matching constraints the approach is shown to allow encodings that provide significantly improved performance over an implementation of a recently proposed predictor from [137] and

encodings assuming correct sequence information.

The remainder of this chapter is structured as follows: first the motivation and approach taken in this chapter to modelling order is discussed in section 6.1. This is followed by a discussion of multiple tree based approaches for realizing the new approach to modelling order in a tractable fashion in section 6.2[1]. In addition to this discussion, section 6.1 briefly reviews some related work and motivates the selection of the Cover Tree as the basic index for the multiple tree approach. The next section provides an introduction to the Cover Tree data structure (section 6.3). The remainder of the chapter is the devoted to the presentation of the new approach, the proposal of novel encodings modelling order and finally experimental results (sections 6.4, 6.5 and 6.6 respectively).

6.1 Modelling order

The observation that sequence can not necessarily be relied on has been made throughout this thesis. Primary motivations come from the noisy nature of GPS data as discussed in chapter 2, section 2.2 which means that missing data points are common. Resampling is one way of addressing this, however, it can introduce many points of error from a single erroneous reading by interpolating out to, and back from, that reading. Additionally it has been noted that the sample rate of devices is generally unknown and not guaranteed to be consistent. This is particularly true across the vast array of different devices that contribute to the historic trails in practice. Therefore using sequence to align two or more movement trails is problematic. Even if resampling is used, and the error this introduced is deemed acceptable, then distance metrics based on simple pairwise comparisons still do not provide an intuitive distance measure with respect to the holistic difference between the trails in many cases (as discussed with, respect to evaluation, in chapter 3).

Despite this current predictive models still typically focus on using sequence as a fundamental structure within the models, matching sequentially in a pairwise fashion. As such time is rarely modelled as feature, but rather implied in the structure of the model. This can be clearly seen in a fixed order Markov Model (such as utilized by [11]) where

[1]Recently, in the time between the submission of this thesis for examination and the completion of corrections, an alternate approach for efficiently calculating the Hausdorrf distance between trails was proposed in [145]. In contrast to the method presented here individual trails are indexed as a logical unit within the tree structure. A comparison of the two methods with respect to performance is interesting future work.

the sequence is encapsulated as the transitions and all other features act to determine the set of distinct states. In universal prediction approaches (see section 2.1.3) sequence is again encapsulated in the transitions. Of note, however, is the approach taken in [137], within the class of pattern-matching algorithms, where the variable length model is extended with timing information in addition to sequence information. Using a standard prefix tree data structure the approach additionally annotates edges with minimum and maximum transition times. This extension provides additional time based discrimination via a penalty function if the timings on the input trail do not fall into the bounds defined at that transition. The approach models time at a very coarse level and as an addition to the sequence information embedded in the underlying model structure. The use of maximum and minimum bounds, however, is clearly only useful if tight bounds exist. It is of note that in this approach time is still modelled separately to the other features (in this case only spatial position), with the structure of the model adjusted to include the information. As such different ways of modelling time require structural changes to the model, making alternative modelling attempts non-trivial. Prediction by principal components has also followed the trend of encoding time only as sequence information, with implementations using a binary encoding scheme to encode features (typically spatial locations) with time used to line up the features [56]. The alignment prediction approach [178] also defines time though sequence as part of the model. In this approach the sequential information is used for matching but in a relaxed way with distance functions used allowing individual points within the two trails being matched to be omitted and a match still made.

A primary motivation for defining time as sequence embedded in the model structure is computational tractability. If sequence is not used as the primary dimension of comparison then calculating the similarity between two trails (E, the input and H, the historic trail under consideration) increases from $max(|E|, |H|)$ to $|E| \times |H|$ in the naive case. With efficient data structures the former is reduced to $|E|$ when quantization and exact matching is used by utilizing a prefix tree data structure. Therefore models range in complexity from $O(|E|)$ to $O(|E| \times |H|)$ depending on how exactly sequence is relaxed and the data structures used. Of all approaches the alignment predictor [178] allows the greatest modeling of sequence, allowing a more relaxed form of sequence matching by allowing gap symbols to be inserted into the sequences being compared. The gap symbols are penalized by a user-defined amount. The method reflects the worst time complexity, $O(|\mathcal{K}|m^2)$ where $|\mathcal{K}|$ is the number of historic observations that the predic-

tor has to reason with (the training set denoted $\mathcal{K}$) and m is the maximum length of the considered sequences. It must be noted, however, that this is a worst case bound under integer programming techniques and heuristics exist to reduce this, with authors reporting that such heuristics still maintain reasonable prediction accuracy [8].

While enabling a relaxed notion of sequence, control over the relaxation via the alignment prediction approach is somewhat unintuitive with regard to movement prediction. Specifically gaps must be assigned a cost, the value of which is left to the implementer. [177, pg. 234] suggests the use of $\frac{1}{r}\sqrt[n]{n}$ as a fixed cost for the distance between a gap and any symbol where n is the dimensionality of the feature space (2 in the case of a Euclidean location feature space) and r is a user-defined parameter for which no further comment is made. In other words the suggestion is that the gap cost should be fixed and equal to a user-defined proportion of the maximum distance possible within the feature space. Since the choice of r clearly affects the performance of the approach, a non-parameterized approach is desirable.

In contrast to previous modelling approaches this chapter presents a novel approach that allows the treatment of time as one of n features (typically one of the features would be the spatial location) in a n dimensional search space with the all points within a trail being connected. Allowing time to be modelled as a design choice in the feature vectors and custom feature distance functions has the following benefits:

- The modelling of time as a feature. Since time is modelled as a feature, its importance can be weighted like any other feature, allowing a user-defined level of trade off between spatial error and temporal error.
- Data-driven gap penalization. Gaps are penalized by the distance to the nearest point within the historic trail, rather than a fixed penalty value.
- Match guarantees. Single points in a historic trail can be part of multiple matches. This prevents excessively long inputs not matching any trail while still providing a penalty mechanism. Additionally it can help prevent excessive penalization of inputs with a higher frequency rate depending on the encoding.
- Tractable complexity and fast empirical performance (see sections 6.4.3 and 6.6).

The approach is realized as a search problem between unordered groups of features in a n dimensional feature space. Computationally prohibitive and intractable for large historic data sets in the naive case, a predictor using multiple iterators over an aug-

mented version of the Cover Tree data structure is proposed. The result is a tractable solution for large data sets with a runtime logarithmic in the number of unique historic points. The approach can be seen to be generalizing existing approaches which use prefix trees. Operating in the continuous feature space, at its core the approach relies on the fast nearest neighbour matching between points a problem that has received significant attention in the literature using tree-based data structures.

6.2 Multiple tree methods

Tree-based indexes utilize the structure within the data itself in order to speed up nearest neighbour search. In cases where matching consists of comparing multiple features (in this case individual points in the trails), where those features must be invariant with respect to some model defined consistency function, multiple tree approaches allow feature matches to be explored simultaneously. In many cases this can lead to greater pruning opportunities. Discussed in-depth in the doctoral thesis of Jeremy Kubica [110], a good example is the location of asteroid tracks. For discussion this problem can be simplified to finding a straight line of points over multiple time steps. A tree is used per time step and points in all trees explored simultaneously to determine a set of points across all time steps that are consistent with a straight line. At each level of the search, the set of selected nodes represent areas of reducing size. To satisfy the consistency function these areas must line up such that a straight line through the regions is possible. As the search descends an individual tree, nodes are pruned if no consistent model is possible. Model consistency is checked by looking at the areas represented by all nodes being considered at that point in time over all iterators. If a node is explored and it is found no consistent solution exists, back tracking is required. Eventually, however, the nodes in the trees only contain one point each, which represents a consistent solution. Other uses of multiple tree methods have included amortizing the cost of searching for individual nearest neighbours for multiple points [20], applying dual trees to 'N-body'-like problems [80] and spatial intersect queries [111] and spatial join queries between spatial objects which are each indexed by a shared or different index (tree) [87, 190].

Multiple trees are also typically used when answering complex queries within databases, where multiple indexes are used. Complex queries are defined as queries in which mul-

tiple similarity (distance) predicates are allowed. A typical example in the literature is the feature *shape* and the predicate *shape = 'round'*. Complex queries were initially proposed considering each index as a black box which provided a sorted list of matches from lowest distance to highest distance, and (optionally) allowed random access [62, 63]. For each predicate, each list of matched objects is read in sequence, forming one set of objects for each predicate until the union of sets contains k objects. Random access is then used to get scores for each object in each set for each predicate and the global score calculated from these scores and the results ranked. Approaches to answering complex queries have since been expanded to take advantage of the indexes structure with, for example, [39, 41] using M-Tree variants as the index.

Framing the movement prediction problem as a complex query one can be done by considering the database to store each trail as one object in the database. A complex query composed of one predicate per point in the input trail is used. The question then becomes *to which feature should each predicate refer to?* This in turn depends how the trail was stored and decomposed into features. If each point in each historic trail was stored as a feature then unordered matching is not possible since each predicate can only refer to one feature. If on the other hand all the points are stored as a bag of points making up a single feature, then complex query mechanism no longer deals with the complexity, instead assuming it is dealt within function used to evaluate the predicate (which using a naive approach returns to comparing the input point with all points in the historic trail). Therefore, while using multiple trees, in this case the trees are used to accelerate search under the assumption of a fixed set of features to which each part of the query is addressed. This is in contrast to the requirement that points in the input be able to match to the best point in each historic trail as defined by the custom distance function.

While the use of multiple trees is not new the novelty lies in the specific framing of the movement prediction problem and the proposal, and evaluation, of a consistency function for this problem that eliminates the need for backtracking ensuring efficient performance. Additionally, in order to realize the proposed consistency function, minor structural modifications to the individual trees is required. The approach uses one tree (in this case each tree is simply an independent iterator over a single tree) per point in the input movement observation. The proposed function and tree modifications are discussed in section 6.4.

The use of multiple trees supposes the choice of a tree based index of which many have been proposed. The most widely used are KD-Trees, QuadTrees, R-Trees and M-Trees along with various variants of each. These are briefly detailed for completeness, followed by a discussion and the motivation of an alternate index with similar properties to the M-Tree, the Cover Tree.

6.2.1 Quadtrees

Quadtrees are a conceptually straight-forward spatial index commonly used to index two- and three-dimensional data. Typically called octrees with respect to three dimensional data they recursively partition the feature space based on fixed regions or centred on a point [167]. In the case where a point is chosen the space represented by the parent is split in two on each dimension centered on the point. In the case of two dimensional data this results in four quadrants, in the case of three dimensional data the result is a partitioning into octants. When fixed-region partitioning is used, square-shaped partitions are formed in the former and cube shaped partitions in the latter. In the case of point based partitioning where the partition shapes are variable, the resulting space partitions can be visualized as rectangles and rectangular prisms of space respectively. Quadtrees in general are not limited to two or three dimensions. However, as the number of dimensions of the data space increases the number of children per node increases exponentially (2^d). This is alleviated by making use of KD-Trees.

6.2.2 KD-Trees

KD-Trees are binary trees, where at each internal node, the space is split into two parts. Traditional implementations of KD-Trees require that the partitions are axis-aligned. While other proposals exist (such as the recent proposal in [96]) the discussion here will be limited to the traditional case with the similarities sufficient for discussion. Construction of a KD-Tree involves the selection of an arbitrary point as the root node, the choice of which affects the index structure and hence performance but not the correctness of the algorithm. Next a point is selected along with a dimension. The value of that dimension for that point then becomes the splitting hyperplane. There are many ways of choosing the point and splitting dimension and hence the splitting hyperplane. Again the choice does not affect the correctness of the algorithm, but only the efficiency. A common

method is to simply cycle through the dimensions and select the point representing the median along the currently selected dimension.

When a balanced KD-Tree is constructed and the points are randomly distributed, Nearest Neighbour search in a KD-Tree has a time complexity of $O(\log N)$ [68]. However if the tree is not balanced and the points are not distributed randomly, then the number of inspection operations can come close to the number of points, n (see [138] for further discussion and examples). In [115] a worst case bound of $O(d \times n^{1-\frac{1}{d}})$ was shown where d is the number of dimensions and n the total number of nodes. In practice, however, KD-Trees often perform very well for low dimensions and approximation algorithms exist resulting in a large variety of uses (e.g. ray tracing [192], shape indexing [15], robot learning [138], image descriptor matching [179]).

Nearest neighbour search is performed by starting at the root of the tree and following the branches depending on which side of the splitting plane the query point lies. Reaching a leaf provides the algorithm with a first approximation of the closest point. Any considered region must now lie within a hypersphere with radius equal to the distance between the query point and the first approximation to be considered. The algorithm then backtracks to the parent and determines if the region represented by the parent's other child intersects the hypersphere. If it does, the algorithm must recursively explore the child. If not, the algorithm backs up to the next parent.

6.2.3 R-Trees

In comparison to KD-Trees, R-Trees [84] represent a bottom up approach, first partitioning the data into groups of small cardinality and then for each group creating a minimal bounding rectangle. Recursively using the new set of rectangles as the data points a tree structure is created. Bounding rectangles can overlap in R-Trees, but not in some variants, such as R+ Trees [172], where the bounding rectangles at the same tree level are not allowed to overlap, giving point query performance benefits as spatial regions are covered by at most one node. Of note is that R-Trees do not have to just store point data, but can store bounding rectangles containing arbitrary shaped data at the base level. Extensively used in practice, poor tree construction can result in poor query runtime performance although recent methods typically result in good real world performance (see [130] for a good discussion of R-Trees and their applications). Of note is a variant called the priority R-Tree [9] which is able to answer window queries with

a complexity of $O((N/B)^{1-\frac{1}{d}} + S/B)$ I/O operations, where N is the number of hyper-rectangles stored in the tree, B is the disk block size, d is the number of dimensions and S is the number of reported rectangles.

6.2.4 Metric Trees

Metric trees [188] represent a class of data structures which address the problem of indexing data in metric spaces rather than vector spaces. A notable consequence is that the data is no longer required to be governed by a L_p distance function such as the Euclidean distance. All that is required is that the distance (dissimilarity) function between two objects is a metric with the data structures using the triangle inequality property to prune the search space[2]. As such the indexing structure is applicable to a wider range of problems where complex distance functions are required to quantify similarities between multi-dimensional features.

Within the class of metric trees *M-Trees* [40] have been proposed which construct a tree in such a way that each node has an associated radius, r for which all the children (recursively) are at most distance r from the feature (data point, e.g. $[x, y, t]$) associated with the node. Utilizing a bottom up approach the algorithm selects points as nodes, associating with them radii of varying sizes depending on the children added to the node. Children are then themselves parents (unless they are leaf nodes which contain the object, or a pointer to the object) with associated radii, with the children's radii being smaller than their parents and describing a region of space contained by their parent. Overlap within the same level is allowed. Inserting a data point into the tree involves recursively descending the M-Tree and locating the *most suitable* leaf node to place the data object in. The criteria of most suitable is in the first instance the leaf that can contain the new point without expanding any radii. If multiple possibilities exist, due to overlap, the leaf whose center point the new point is closest to is heuristically selected. If the point does not fall within any existing leaf radii, then an attempt is made to expand a node's radius. If the expansion would result in to many points being placed in a single leaf then the leaf is split. The exact procedure for the last two steps is heuristically driven. If this heuristic performs poorly then a large amount of overlap

[2]Triangle inequality states that, given any three points (A, B, C) and the pairwise distances between them ($|AB| = x,\ |AC| = y,\ |BC| = z$), the sum of any two distances is greater than the third distance, see chapter 1 in [105] for an in-depth explanation

can occur, severely degrading query performance.

6.2.5 Discussion of tree-based indexes

In the preceding sections a number of tree-based indexes were discussed which potentially can form the base tree used within a multiple iterator approach. It is of note that all of these could be used. Despite this an alternative tree-based index, the Cover Tree is chosen. The departure from standard indices such as the R-Tree or the KD-Tree is motivated by two factors:

- Movement prediction can potentially benefit from using an array of contextual features (such as weather, mood, social context, etc.) that are not well represented as numerical values in a multidimensional space, but for which a complex distance function can be generated. Cover Trees, like M-trees, are hence favourable because they assume a Metric and not a vector space.
- It is envisioned that a large number of features will be incorporated, and hence structures that are designed to perform well with high dimensional features is desired. Unlike Cover Trees, the M-Tree and its variants are not guaranteed to have logarithmic runtime performance in the number of points, and thus the former is preferred.

In closing it is important to note that the choice of the underlying tree structure only influences the runtime performance of the predictors examined in this chapter. The performance, in terms of prediction accuracy, of the predictors due to the proposed approaches to modelling order is completely independent of this choice. The comparison of different underlying indexes, with respect to runtime efficiency, has the potential to make interesting future work.

6.3 An introduction to Cover Trees

Having motivated the use of multiple trees and the selection of the Cover Tree as the basic data structure an introduction to this data structure is provided in this section. Directly following this the augmented version is proposed detailing the multiple iterator approach, the proposed consistency function and the minor structural modifications.

Cover trees provide a tree-based data structure for fast nearest neighbour operations in metric spaces where the data are points within the space [21]. A cover tree has a space complexity of $O(n)$ and a nearest neighbour complexity bounded by $O(c^{12} \log n)$ where c is an expansion constant which is small in datasets with low intrinsic dimensions and n is the number of indexed points. As such, the approach is *guaranteed* to be logarithmic in n. In [108] cover trees were shown to outperform a naive brute force approach at nearest neighbour queries with as little as 300 training points on a uniformly distributed synthetic data set with the Euclidean distance metric. The performance in practice with respect to movement prediction is examined in section 6.6.2.

6.3.1 Preliminaries

Cover trees index points within a *metric space*. In contrast to indices in vector spaces (e.g. KD-Trees) cover trees only require that a distance function be specified that operates over the points, and that this distance function is a *metric*. Given the set of points to be indexed $p_0, \ldots, p_n \in P$, in order to qualify as a metric, the distance function $\delta(p_t, p_s)\ \forall t, s$ must satisfy the following properties:

$$\begin{aligned}
&\delta(p_t, p_t) = 0 && \forall t && \text{(reflexivity)} \\
&\delta(p_t, p_s) > 0 && \forall t \neq s && \text{(positivity)} \\
&\delta(p_t, p_s) = \delta(p_t, p_s) && \forall t, s && \text{(symmetry)} \\
&\delta(p_t, p_s) \leq \delta(p_t, p_k) + \delta(p_k, p_s) && \forall t, s, k && \text{(triangle inequality)}
\end{aligned}$$

6.3.2 Cover tree data structure

A cover tree is a tree-based data structure constructed such that at each level of the tree the nodes in that level are all points that have a distance greater than 2^i between each other. This is called the separation property. Stepping down a level in the tree the children for each node include the point represented by the parent (nesting property) and have the property of being separated by a distance of at least 2^{i-1} with respect to each other (separation property at the next level down) while being within a distance of 2^i with only one point at the ith level, its parent (covering tree property). As such only one path exists to each node in the tree (down from each child's unique parent) with the parents being close to the children. This is shown graphically in figure 6.1. Formally:

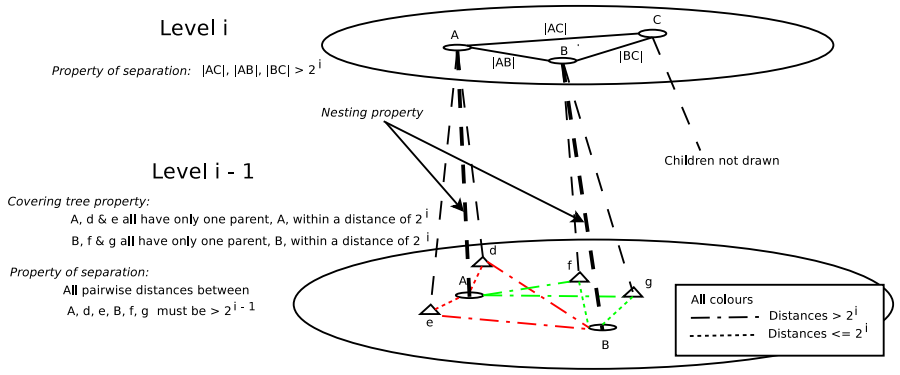


Figure 6.1: A graphical representation of the Cover Tree properties

Definition 1. *A Cover Tree T on a dataset $p_0, \ldots, p_n \in P$ has the following properties:*
Let i equal an arbitrary level in the tree.
Let L_i represent the set of points in the tree at level i.
Then:

$$L_i \subset L_{i-1} \qquad \textit{(nesting property)}$$

$$\forall p_t \in L_{i-1},\ \exists! p_s \in L_i : \delta(p_t, p_s) \leq 2^i \qquad \textit{(covering tree property)}$$

$$\forall p_t, ps \in L_i, \delta(p_t, p_s) > 2^i \qquad \textit{(separation property)}$$

Considering extremes, when i is sufficiently large ($i \to \infty$) the distance that each point in the current level must differ by to be included is so high that only one point can exist. Conversely, when i is sufficiently low ($i \to -\infty$) then the distance that each point has to be within is sufficiently low and all data points are present as unique nodes.

Intuitively, at the higher levels of the tree only a few, spread out, data points are visible. As one proceeds to lower levels of the tree more and more points come into view, dispersed relative to the minimum inter-point distance permitted at that particular level.

6.3.3 Constructing a Cover Tree

A cover tree can either be constructed by a set of consecutive inserts or using a batch construction algorithm [20]. Theoretically the batch construction algorithm has the same guarantees as building by consecutive inserts. However, batch construction is expected to be much faster since it is able to amortize the construction over multiple points[3], with

[3]In other words rather than starting at the top of the tree each time for each point points can be added along the way distributing the cost of the decent over multiple points.

initial studies confirming this in practice. Here only the consecutive insert procedure will be detailed.

The insert procedure starts by assuming a root node. If one does not exist then one is created at the theoretical ∞ level, which in practice corresponds to a level i which when used as the exponential to 2 provides a radius around the root node that encompasses all possible points. Subsequent insertions into the cover tree then starts with this root node. Assuming a root node does exist, the algorithm starts at the root. If the new point is within 2^i of this root node and more than 2^i away from the roots children then the root is selected as the new points parent. If not then the procedure is applied to all children of the root which are within a distance of 2^i. Due to the nesting property, even if only the root exists in the tree, there is a child of the root (once again the root) at the next level. Therefore, eventually, there will be a level for which the new point can be inserted. The psuedocode is shown as algorithm 1. It has been shown that inserts take time at most of $O(c^6 \log n)$ [21]. Rather than storing a copy of the node each time it is inserted, resulting in a large amount of duplication due to the nesting property, the data structure can use a representation of the tree that stores each point in tree once and a pointer to its children at each level. As such the cover tree only requires $O(n)$ space.

Algorithm 1 Cover tree insert method [108], corrected from [21].

Insert(point p, cover set Q_i, level i)

$Q = \{Children(q) : q \in Q_i\}$
if $\delta(p, Q) > 2^i$ **then**
  return "parent found" - True
else
  $Q_{i-1} = \{q \in Q : \delta(p, q) \leq 2^i\}$
  found = Insert$(p, Q_{i-1}, i-1)$
  if found and $\delta(p, Q_i) \leq 2^i$ **then**
    pick a single $q \in Q_i$ such that $\delta(p, q) \leq 2^i$
    insert p into $Children(q)$
    return "finished" - False
  else
    return found
  end if
end if

6.3.4 Nearest Neighbour queries

Searching for the nearest neighbour in a cover tree works by first calculating the distance between the point and the root, α. It can be shown (see [21] for the proof) that the closest point must either be the root, or be a point at the current level within α, or a child of a point at the current level within $\alpha + 2^i$ where i is the current level. Therefore the search progresses by, at each level, calculating α as the minimum distance between the query point and any node under consideration at that level and then selecting all points within $\alpha + 2^i$. The children of these points (remembering each parent is a child of itself) then become the nodes under consideration at the next level. As $i \to -\infty$, $2^i \to 0$ and only a single point is left. The algorithm is detailed in algorithm 2.

Algorithm 2 Cover tree method to find the nearest neighbour [21].

Find-Nearest(cover tree T, query point p)

Set $Q_\infty = C_\infty$, where C_∞ is the root level of T.

for $i = \infty$ to $-\infty$ **do**

Set $Q = \{Children(q) : q \in Q_i\}$

Form cover set $Q_{i-1} = \{q \in Q : \delta(p, q) \leq \delta(p, Q) + 2^i\}$

end for

return $\arg\min_{q \in Q_{-\infty}} \delta(p, q)$

6.3.5 Other cover tree operations

In addition to insert and nearest neighbour methods, point deletion, batch construct and batch query methods have been proposed [20]. These are not detailed here as they are not required for movement prediction in the basic case. Their extension in the case of the augmented structure is relatively straightforward and their exact specification left as future work. Of note is the batch query algorithm which should enable the reduction of the theoretical complexity of finding nearest neighbours simultaneously, amortizing the descent of the search across all queries resulting in a theoretical bound $O(c^{16} \log n)$ for any number of queries. For a small number of queries, as can be typical in movement prediction since each query represents one observation of the moving object so far, the method may or may not perform better and as such the utilization of this algorithm requires an in-depth comparison before its wholesale use.

6.3.6 Summary

The cover tree provides a tree-based data structure with known time complexities logarithmic in the number of data points. Providing fast range queries for individual points, the structure can be leveraged to enable fast, nearest neighbour queries between movement trails, as will be detailed in the following section. The approach indexes the very general metric space where only a distance function where triangle inequality holds is required. Additionally designed to perform well under high dimensions, the data structure is able to generalize beyond the small number of features investigated in this work enabling a large number of features to be encoded potentially providing better prediction accuracy and/or to address the more general problem of context prediction.

6.4 Augmented Cover Trees

In this section an augmented Cover Tree is presented and an algorithm given to efficiently perform nearest neighbour queries where the query is a group of unordered feature points in an arbitrary multi-dimensional feature space.

6.4.1 Problem definition

The augmented Cover Tree allows the arbitrary specification of features and an arbitrary distance function between points. Compared to the traditional cover tree the augmented cover tree additionally stores a set of labels identifying the trails to which the point and any of its children belong. This data is used along with multiple iterators in a new method for finding the closest trail, which is a set of points, rather than the closest single point or the closest (possibly different) trail for each point. Therefore the augmented cover tree provides a new structure and the mechanisms required to answer the queries required for the task of movement prediction. In contrast, traditional cover trees can only answer queries about single points.

Given a set of pairwise distances between all unordered features in two trails a global function is required to arrive at a final distance score between the trails. Note that this global trajectory distance function may be asymmetric. Having a non-metric similarity function is actually very appealing in this problem-space, because it reflects the notion that an input trail is only meant to match part of the historic trail and not vice-versa.

This function is equivalent to the consistency function used by [110] when using multiple trees. This function is tied into the data structure and corresponding algorithm and not easily changed. In the next sections two global functions are considered.

In both cases when considering formal definitions the following symbols will be used as defined, considering feature points in an arbitrary multi-dimensional feature space.

- Let a historical trail H, be encoded as a set of feature points, $\{h_0, \ldots, h_{|H|}\}$.
- Let the input query trail, E, also be a set of feature points, $\{e_0, \ldots, e_{|E|}\}$.
- Let $\delta(x, y)$ be a dissimilarity (distance) function that provides a numeric level of dissimilarity between two arbitrary feature points.

Nearest unordered group minimizing total distance between points

Formally the problem of determining the nearest unordered group minimizing total distance between points is:

$$TMC(H, Q) = \sum_{e \in E} \min_{h \in H} \delta(h, e) \tag{6.1}$$

Hence, for each point, e, in the input trail, E, the distance is found to the closest point, h, in the historical trail, H, and finally sum the results. This approach is perhaps the most intuitive when considering nearest unordered group matching.

Nearest unordered group within a minimal radius of each input point

This criteria corresponds to the Hausdorff distance. The Hausdorff distance was previously discussed in chapter 3 with regard to trail comparison. In that discussion the truncated average discrete Fréchet was preferred. Here, however, the computation speed of the comparison is important since real-time prediction is desired. This is in contrast to the less time critical process of prediction method evaluation. Therefore, in this work the Hausdorff distance is considered due to its ability to be efficiently computed using the augmented Cover Tree data structure presented within this chapter. Note also that in the case of matching for prediction asymetric comparison is allowed, which makes the Hausdorff distance directly applicable.

In contrast to the previous nearest neighbour criteria, this criteria enforces the requirement that all input points must match, in an unordered fashion, equally well to the

historic pattern. The worst point matched is then used as the score from which the historic patterns are ranked. From this the lowest scoring is selected as the prediction. The requirement skews the choice of patterns based on the most different point. In situations considered to have limited symbol corruption noise[4] this may be of benefit if small deviations in the historic patterns are what discriminate between them. An example highlighting this is shown in figure 6.2 (a). In noisy environments, however, the reliance on a single point can artificially exclude patterns. This is highlighted in figure 6.2 (b). Empirically, however, it is of note that the Hausdorff distance criteria shows the best performance (as will be shown later in section 6.6.3). It is the Hausdorff distance criteria (HDC) for which the augmented Cover Tree has been developed. Importantly the Hausdorff distance criteria only requires a weak form of back-tracking up the tree. Specifically to determine the closest trail to the input only the continual decent of the tree is required. The descent continuously, globally, restricts the minimum radius. It is of note that the augmented Cover Tree could be used to calculate the TMC, since the HDC is an upper bound on the distance. This, however, would involve back-tracking. It is of note, however, that this is not investigated or implemented in this thesis and other data structures may be better suited to this global criteria. Instead the TMC is implemented, purely for comparative evaluation, in a naive fashion using a brute force approach where the input is compared directly to each historic trail and the total minimum distance calculated.

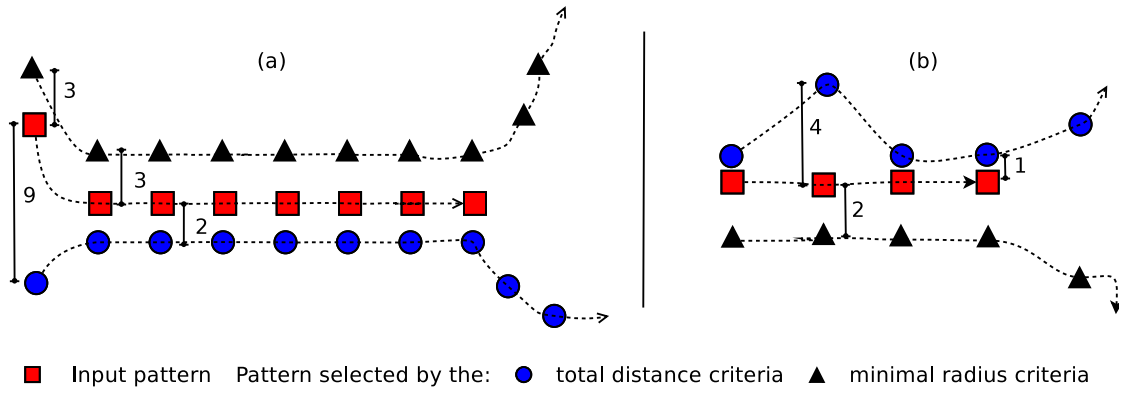


Figure 6.2: Examples where the Hausdorff distance criteria would select the intuitive solution (a) and vice versa (b)

Formally the problem of determining the nearest unordered group within a minimal

[4]Filters can be employed to drop uncertain or impossible samples using domain knowledge either in at the device level or in a preprocessing step to reduce symbol corruption, although this is not done in this work.

radius of each input point is:

$$HDC(H, E) = \max_{e \in E} \left(\min_{h \in H} \delta(h, e) \right) \tag{6.2}$$

6.4.2 Augmented cover tree data structure

In order to answer nearest neighbour trail queries efficiently the cover tree data structure is modified. The primary modification is the addition of a set of trail labels to each node containing a unique list of the trails (historic observations) that the node and its children belong to. Utilizing this label information nearest neighbour trail queries can be performed by a multiple iterator approach over the tree, with one iterator per point in the input.

Algorithms for nearest neighbours via the Hausdorff distance criteria

The pseudocode for nearest neighbours using the Hausdorff distance criteria and the augmented Cover Tree is shown in algorithm 3. The algorithm starts by instantiating one iterator per point in the input over the cover tree, starting at the root node. Additionally the initial (global) label set is set to all possible candidate trails, which are all trails for which enough length exists to form a prediction after the input has been matched to the historic instance[5]. At each step all iterators recurse down a level of the tree in parallel, individually inspecting the children of all points declared in range in the previous iteration and forming the next set of points (the coverset) that must be inspected based on the known upper bound of the minimum distance. This is defined as the global maximum of the distance between the input point and the closest point in each iterator plus a decreasing bound based on the granularity provided by the Cover Tree at the specific level. Note that in contrast to the standard case, this upper bound is based on a global bound over all points (line 14). The label set is then computed for the new coverset indicating the observations which have at least one point supporting them in each iterator (line 16). In this way the recursion simultaneously restricts the area of space being investigated for close points around each input point and reveals additional points within the area, further constricting the region under investigation if closer points are found.

[5]It is of note that the construction of this set can be delayed assuming it will be significantly smaller after traversing the tree somewhat. This heuristic then potentially requires the algorithm to backtrack if the set is not constructed soon enough and it is found to be empty upon construction.

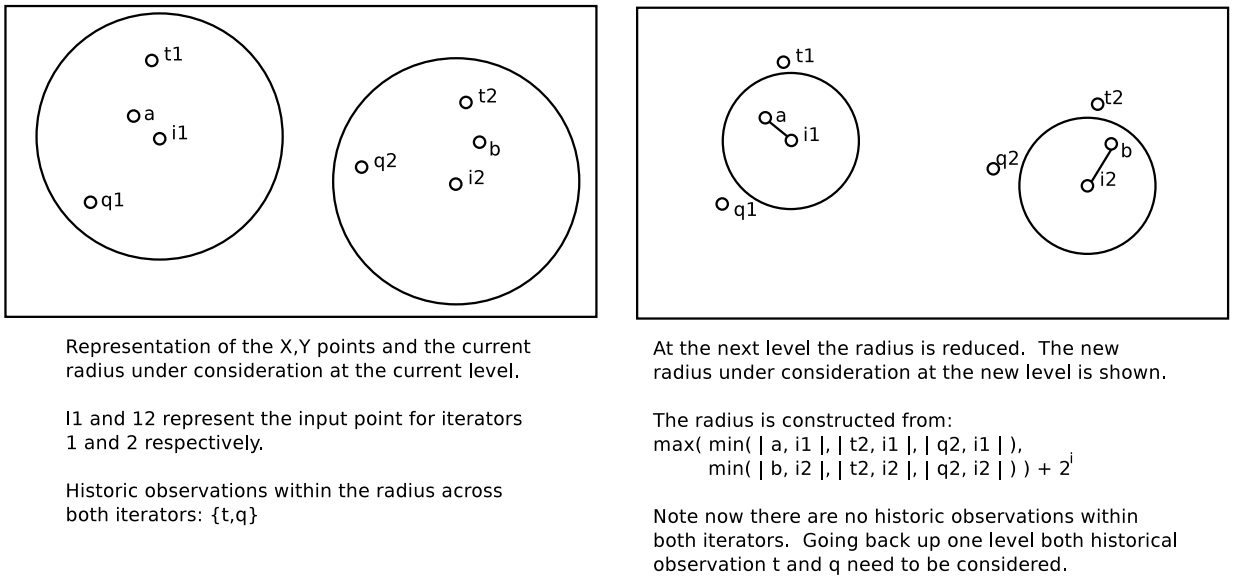


Figure 6.3: Example illustrating that multiple historic observations can be returned by the cover tree matching.

In the case of a single point the systematic reduction of i to $-\infty$ will eventually separate out the nearest point. However, with multiple iterators, it may be that multiple points are left in the space bounded by the radii selected at each level. This is because each iterator acts locally when recursing down the tree at each level, with the global condition checked after descending to the next level. This is shown visually in figure 6.3.

It is possible to envision an algorithm that avoids this. For example the point that contributed the minimum could be removed from consideration and a new global minimum obtained with no additional distance computations, repeating the process until the label set was non-empty. As such no back-tracking up the tree is required, at the expense of a more complex function to determine the next minimum distance that will be applied globally. Of note is that the function would not require any additional distance calculations. However, the set of observations here is typically very small[6] and so in practice, in the implementation used in thesis, the final solution is solved via a brute force calculation.

The impact of the Hausdorff distance criteria on the algorithm is that it is not always necessary to proceed to the full depth of the tree as the termination condition is dependent on the label set. Therefore after each recursive step over all input points, the global

[6]Average 2.3 on a random subset of 102 tests with 58% of the time only one trail returned, 93% fewer than 5. Only 2 returned more than 10. The most returned was 23 and the size of the training set was 918 historic trails.

label set is checked, if it contains only one trail that is returned. If it empty then no historic trails exist within the specified Hausdorff distance upper bound and the previous label set is returned. The final case for termination is the standard one, the bottom of the tree is reached and the result, or in this case the label set, is returned.

The proof that the algorithm will indeed return the Hausdorff distance matched trail follows directly from the proof of the nearest neighbour of a point from [20, pg. 4 Theorem 2.2]. Specifically, for each iterator, for any point q in C_{i-1} the distance between q and any descendant q' is bounded by $d(q, q') \leq \sum_{j=i-1}^{-\inf} 2^j = 2^i$. Consequently the step in line 13 can never throw away a point that is within the global minimum distance. Eventually one of the termination conditions detailed in the above paragraph occurs, namely either the intersection of the label set will be zero or one, or there are no descendants of Q_i not in Q_i (or in practice we have hit the bottom of the finite tree) and the nearest neighbour must be in Q_i.

6.4.3 Complexity analysis

The augmented Cover Tree utilizes the same representation as the normal Cover Tree, using multiple iterators. As such the $O(n)$ space complexity is maintained, where n is the number of unique points in all historic trails[7]. The time complexity bounds (in terms of the number of distance comparisons) is k times the original time complexities, where k is the number of points in the input. This results in $O(kc^{16} \log(n))$, where c is the expansion constant detailed in [21].

6.5 Models using Cover Trees

Within augmented cover trees a wide range of feature encodings, which *optionally* include time/sequence information, can be used along with custom distance functions. Within this chapter, four different encodings are examined, although many more are possible.

1. **XY:** No encoding of time. Only two-dimensional spatial coding is used. The standard L_2 distance metric is used.

[7]Depending on the use of spatial quantization or otherwise the number of unique points may be equal to or less than the number of points in all trails.

Algorithm 3 Method to find groups with all points within a minimal radius.

Find-Nearest-Groups-Within-Min-Radius(cover tree T, query point set $P = p_0, \ldots, p_n$)

{Initialization}

for $p_k \in P$ **do**

Set $Q_\infty^k = C_\infty$, where C_∞ is the root level of T.

end for

Form initial label set $L_\infty = lengthFilter(k)$

{*lengthFilter* is a function that selects only that observations that have enough length to offer a prediction based on the input length}

{Iterate down tree}

for $i = \infty, \ldots, -\infty$ **do**

$L_{i-1} = L_i$

for $p_k \in P$ **do**

Set $T_i^k = \{Children(q) : q \in Q_i^k\}$

Form cover set $Q_{i-1}^k = \{q \in T_i^k : \delta(p, q) \leq \max_{a \in P} \left[\delta(a, T_i^k)\right] + 2^i\}$

Form label set $L_{i-1} = \{L_{i-1} \cap labels(Q_{i-1}^k)\}$

end for

{If no labels appear in all iterators}

if $L_{i-1} == \emptyset$ **then**

{Return groups at the higher level where labels did appear in all iterators, and that level}

return (L_i, i)

end if

end for

{Bottom of tree reached}

return $(L_{-\infty}, -\infty)$

2. **XY-SEQ:** Strict encoding of sequence. Two-dimensional spatial coding is used, along with a third sequence indicator. A custom distance function is used penalizing the alignments for which the sequence numbers do not match, effectively enforcing sequence. When the sequence numbers match the standard L_2 distance is used.

3. **XY-START:** No time encoding, start point is emphasised via encoding the change in distance from the start point as a feature. Additionally two-dimensional spatial encoding is used. The distance function was the L_2 across all three features.

4. **XY-TD** Relaxed encoding of sequence by encoding the total distance travelled in conjunction with a two-dimensional spatial encoding. The standard L_2 distance metric is used.

The first is the most basic case and neglects to encode time completely. This allows points to be matched completely out-of-order. When the total minimum distance global criteria is used the second represents the type of encoding found in the literature where the sequential structure is used within the algorithms themselves. Differing from previous approaches (in particular the approach from [137] which is later used as the baseline in the evaluation) the distance function is not normalized in this encoding and the matching function is performed as a completely separate step as motivated by the results from chapter 5. The third encoding encapsulates the notion that the start point of a trail is discriminative. A motivating example is someone living on a major road joining two major destinations on the other side of a town. If order is unimportant then the trail that starts halfway along the road and proceeds along it for a number of samples is quickly matched to trails that travel along the whole route to the other side of town, even though in every case the person who lives at this house only drives along the road for ten blocks then turns off at the local supermarket. Therefore the encoding relates to the notion that the start location is significant so trails that start in one location should only exactly match other trails that start at the same location. The final encoding relates to a relaxed notion of sequence where the total distance along the trail is used as a feature.

The encodings detailed above show a number of different ways to model sequence rather simply assuming its accuracy. The difference in these encodings demonstrates the main aim of this chapter, to propose the modelling of sequence rather than assuming correct sequence information, having additionally provided the augmented cover tree algorithm

enabling their efficient use. As is shown in the next section, this modelling extension with the right encoding can lead to increase prediction accuracy, providing state-of-the-art results.

6.6 Movement Prediction using the Augmented Cover Tree: Experimental Results

Movement prediction using the augmented cover tree employs nearest neighbour search. Following this, if more than one candidate prediction sequence exists, the probabilities of the prediction sequences conditional on the input sequence is used to distinguish the most likely prediction. This involves the pre-calculation of support values for each historic trail. These pre-computed values are then used to quickly generate conditional probability scores via lookup. In order to determine the support (done at the same time as the tree construction) the same global matching criteria is used, in the case of the augmented cover tree approaches this is the Hausdorff distance criteria.

To evaluate the cover tree approach, including the use of the Hausdorff distance global criteria and all different encodings, the predictors listed bellow were compared in addition with a baseline method implemented from [137]. Since the method from [137] is parameterized and good values for these parameters unknown, the parameters are chosen experimentally using the same data set. This ensures that any performance increases in the new predictors are not simply due to an under-performing baseline based on poorly selected parameters. In other words, the accuracy of the baseline potentially is somewhat inflated compared to what would be achieved if the parameters were selected based on an independent data set as is considered best practice when evaluating parameterized algorithms. However, considering this method is used as a baseline the potential over-performance of this predictor is acceptable. Specifically the parameters the algorithm requires is a temporal weighting, a spatial weighting and a temporal threshold. The temporal threshold rejects an exact spatial match if the temporal distance is greater than the threshold. This parameter is fixed to six seconds since this is the value used to convert the continuous GPS data streams into trails. The temporal and spatial weighting parameters are then experimentally determined. A parameter plot is shown in figure 6.4. Investigated were the values $0.0, 0.25, 0.5, 0.75, 1.0$ for temporal weighting and $0.25, 0.5, 0.75, 1.0$ for spatial weighting. Spatial weighting was not set to 0.0 since

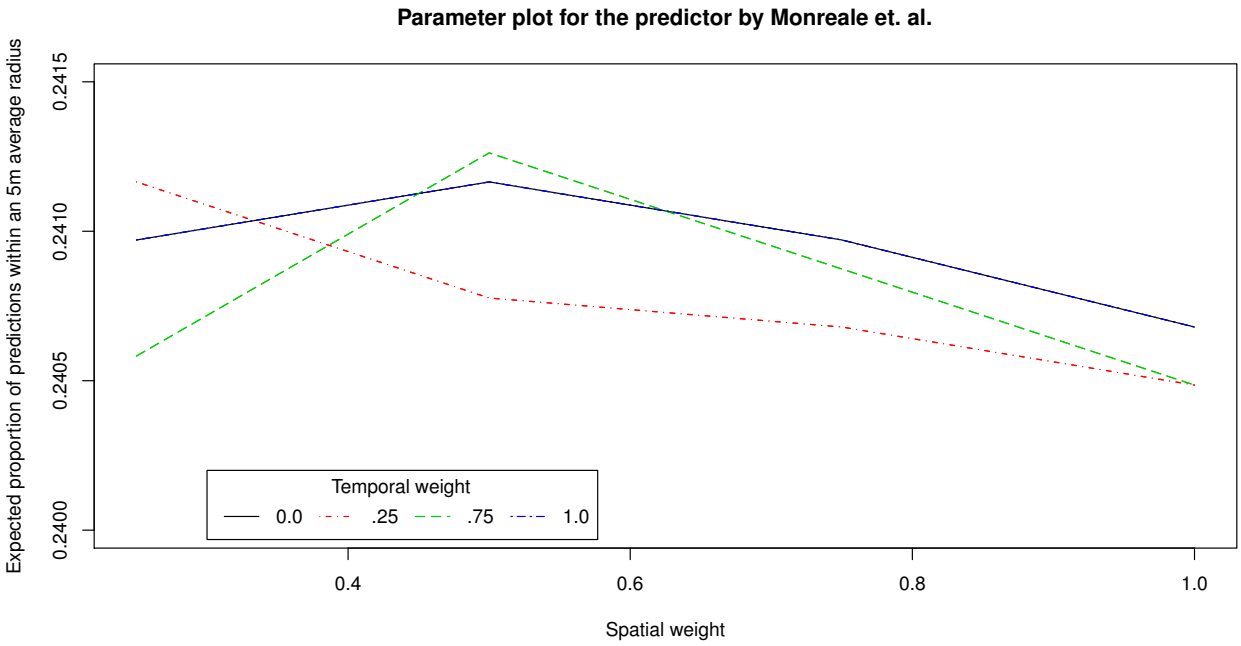


Figure 6.4: Graph showing the mean proportion of routes predicted by the predictor [137] within an average Fréchet distance of 5 metres using different spatial and temporal weights.

this results in almost all historic trails matching making the method intractable. From the results the best parameters (temporal weight of 0.75 and spatial weight of 0.5) were selected.

In total eight different predictor variants were evaluated along with the baseline, the predictor from [137]. These are listed below with the encoding abbreviations as used in section 6.5.

1. **CT:** Cover Tree (Hausdorff distance global criteria)
   (a) XY
   (b) XY-SEQ
   (c) XY-START
   (d) XY-TD
2. **TMC:** Total minimum distance global criteria (via brute force algorithm, not suitable for realtime performance in large data sets)
   (a) XY
   (b) XY-SEQ

(c) XY-START

(d) XY-TD

3. **WN** The predictor from [137] with parameters selected for optimal performance on this data set. The parameters selected were: *temporal weighting*= 0.75, *spatial weighting*= 0.5, *temporal threshold*= 6 *seconds*.

6.6.1 Experimental Methodology

The experimental methodology is based on the recommendations made in chapter 3. In summary, the average Fréchet distance is used as the test statistic for a single prediction and repeated 10-fold cross validation is used for the testing methodology, with 10 repeats. The results are then graphed, displaying the expected proportion of correct predictions with respect to a varying relevance parameter. Side-by-side histograms are plotted of the individual predictions and side-by-side box plots are provided displaying the aggregated results. The value of 5 meters is chosen as the value for the relevance parameter for statistical analysis. For statistical analysis paired t-tests modified for use with the OET estimator of variance (M set to 7) is used with p-values modified according to Holm's procedure since multiple predictors are compared.

The dataset used in the evaluation was the *D-SCENT* dataset discussed in section 2.2.2. For this experiment each individuals' path was broken into trails, by splitting the GPS stream when a gap of six seconds or more occurred. Trails were then only kept if they were at least 20 meters in length and had at least five points. In total 1031 individual trails were created with an average length of 128.7 meters. No spatial quantization was performed, however, the longitude and latitude readings were converted into OSGB (Ordnance Survey of Great Britain) grid references using Jcoord for Java[8].

6.6.2 Runtime performance results

Analysis of the runtime performance is primarily measured with respect to the number of distance calculations as it provides an insight into the performance of the algorithm abstracted from the specific implementation. In addition the number of intersect operations are reported, although it must be noted that the number of intersect calculations

[8]Available from: `http://www.jstott.me.uk/jcoord/`

were not optimized[9] in the implementation and as such represent an upper bound. In general these are expected to be cheaper than the distance calculations. Only the prediction phase is measured. This reflects the assumption that training, in other words building the tree, would be performed offline. As a baseline a brute force approach was implemented using the Hausdorff distance global criteria. In this algorithm each point in the input was compared to each point in each historic path and the maximum, minimum distance for each point in the input used as the distance for the (input, historic) pair.

Performance measurements were taken by withholding a test set of 100 observations and then randomly selecting a training set without replacement of sizes 100 to 900 in increments of 100. For each training set size the resampling was repeated 10 times and the average taken. Finally an overall average for each training set size was taken. It is of note that the number of distance calculations recorded from the augmented cover tree represent the amount when distance caching is used per prediction. The caching is strictly per prediction with all caches being cleared as the first step of each prediction. The baseline brute force algorithm never repeats a distance calculation per prediction and therefore no cache is implemented.

The results are presented in figures 6.5 and 6.6. Figure 6.5 shows a comparison of the number of distance calculations (distance and intersect calculations in the case of the cover tree) made on average per observation for both the brute force approach and the cover tree, clearly showing the benefit of using the tree based data structure. Zooming in on only the cover tree performance (figure 6.6) the logarithmic nature can be seen, but as expected it is converging slowly due to the large constant.

6.6.3 Prediction accuracy results

The prediction accuracy is evaluated using 10×10-fold cross-validation, using the Fréchet distance as the test statistic for a single prediction. The error function (as motivated in section 3.1.9) then converts the Fréchet distances into a proportion based on a user-defined level of an acceptable distance. In this section the parameter is set to 5 me-

[9]Intersects were calculated at each level, even if there was no change in any cover sets. This did not influence the distance calculation count as distance calculations involved were simply repeats of previously calculated distances and hence were taken from the cache and did not contribute to the count of distance calculations.

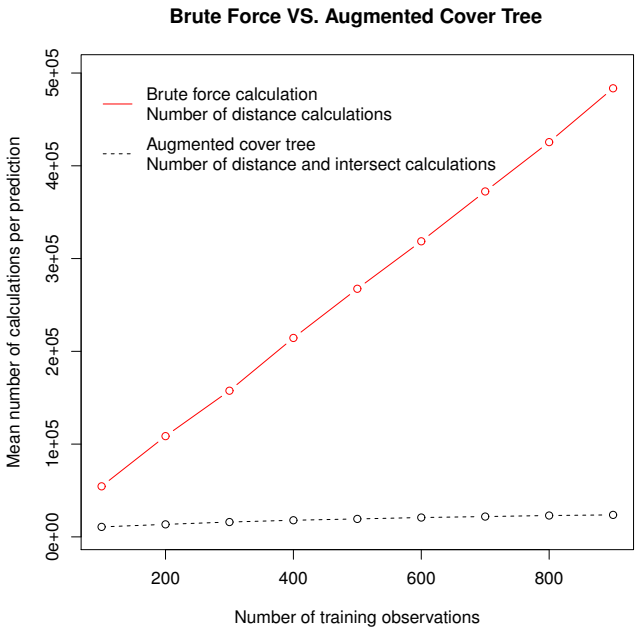


Figure 6.5: Graph showing relative performance (average number of operations per prediction) between the brute force and the augmented cover tree approach to prediction, under the Hausdorff distance global criteria, in terms of the number of distance and distance + intersect operations performed respectively.

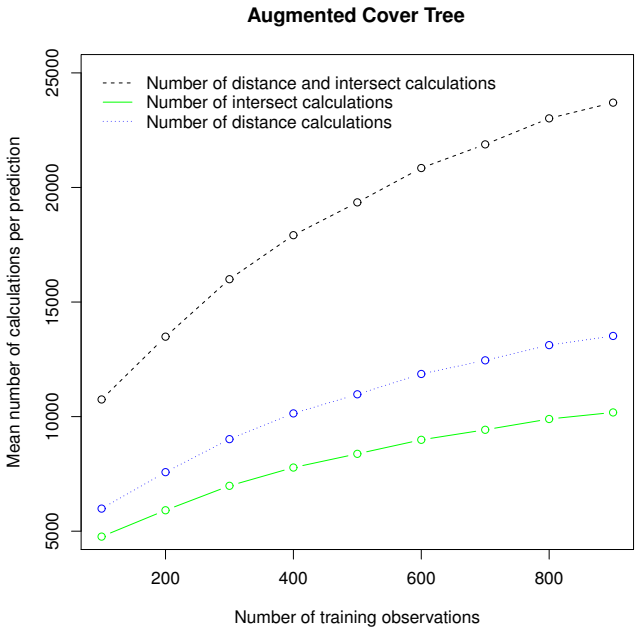


Figure 6.6: Graph showing average number of operations per prediction of the augmented cover tree.

ters. Five meters is selected as this was the level of granularity chosen in the initial development of the DSCENT software which occurred independently of this investigation and represents the initial, theoretically-chosen, level of granularity. Despite having pre-selected a level for further investigation a range of parameter options are still first examined and a line graph plotting the parameter vs. performance is shown in figure 6.7. The graph paints a broad picture of each of the predictor's performance.

A first point of interest is the poor performance exhibited by the baseline predictor. Comparing the baseline to the somewhat similar TMC:XY-SEQ it is of note that the latter shows much better performance. While both assume correct sequence information the latter does not match based on a normalize the distance measure and does not integrate frequency counts of the historic path as part of the matching strategy. Rather, the matching is performed first and the frequency information used if more than one prediction is returned.

The graph also shows the general superiority of encodings that relaxes time but include, as an additional feature, either the total distance travelled so far or the distance to the start point from each subsequent point. note, however, that this property does not emerge until the admissible precision is set to three metres or greater.

Within the better performing encodings the Hausdorff distance global criteria shows better performance. This applies more generally once the admissible precision is set to 5 metres or greater with all encodings showing either equal or better performance than under the total minimum distance global criteria.

In figure 6.8 histograms of the individual prediction scores are shown, up to a Fréchet distance of fifty meters. The graph shows the distribution more clearly, with all methods following a similar overall distribution, with predictions clustered between 0 and 15 meters followed by the remainder of the predictions falling, with a slowly decreasing quantity, over a larger range. Other observations, similar to those from figure 6.7 can also be made, for example the superiority of the CT:XY-TD method, although they are generally less clear from this graph.

The case where the relevance parameter is set to 5 meters is now considered. Figure 6.9 shows a side-by-side box plot of each fold proportion. From the plot it is clear that statistically significant differences exist, at least between the WN and XY encodings and other encodings. Of additional interest is the comparison of the encodings that respect sequential encoding of time, TMC:XY-SEQ and CT:XY-SEQ, and the other methods

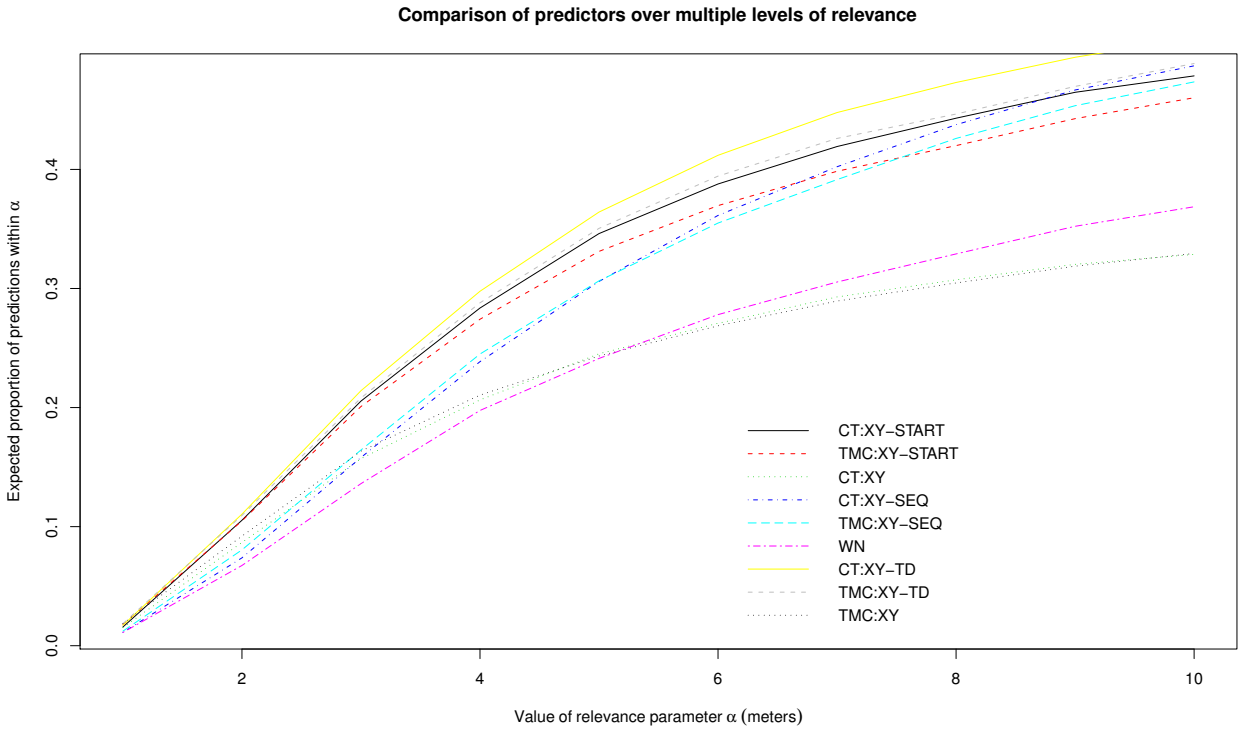


Figure 6.7: Graph showing the performance of the predictors for different levels of the relevance parameter α.

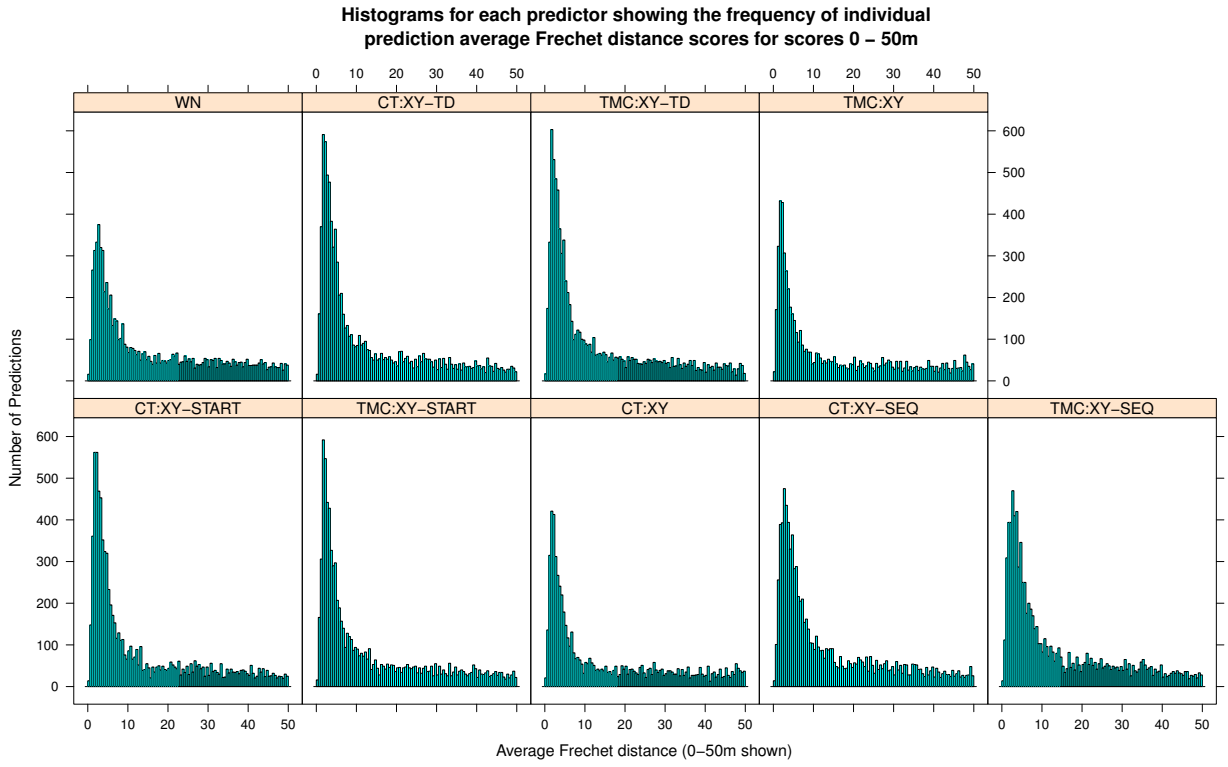


Figure 6.8: Graph showing side-by-side histograms of the individual predictions for each predictor.

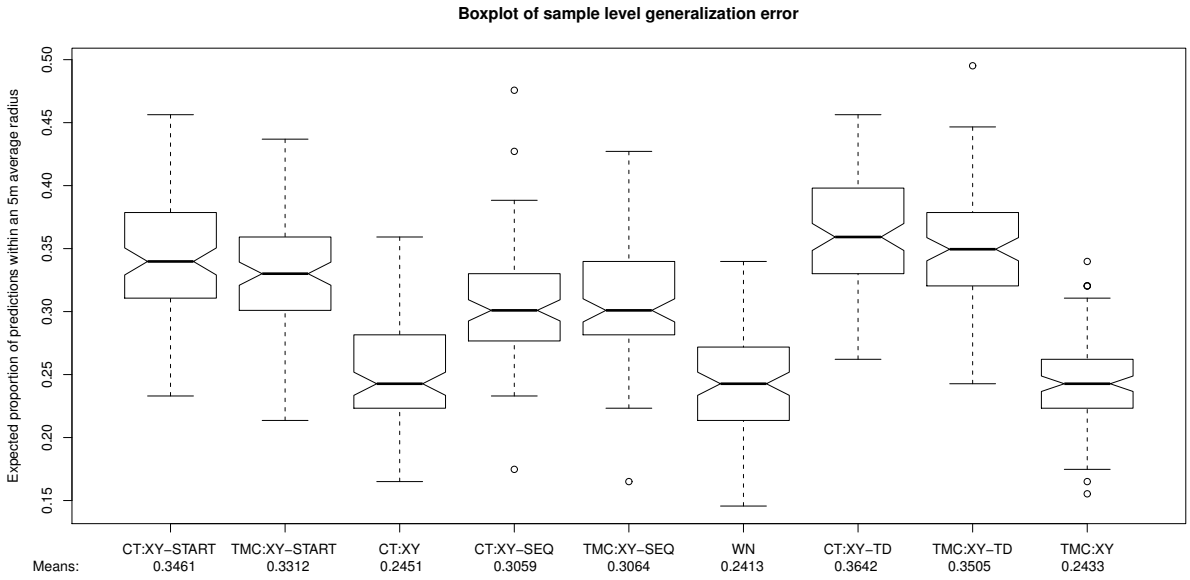


Figure 6.9: Side-by-side box plots showing the proportions from each fold in the cross-validation runs. This represents the sample level generalization error.

(excluding WN and XY) that do not, which also show signs of the existence of a significant difference. The plot does not seem to indicate large differences between the two global distance criteria providing rough conclusions similar to the previous graphs.

Following the methodology outlined in chapter 3 paired t-tests are performed using the conservative OET[10] variance estimator. In calculating the conservative estimator $M = 7$ was used and 10×10-cross validation was used to calculate the means of each split. The p-values are then adjusted via the Holm procedure since it is easily available in the R package *muToss* [140]. Before performing the statistical tests the distribution of each predictor is checked to ensure it satisfies the assumption of normality. This is done via a set of normal probability plots and the D'Agostino test of normality. These are shown in figure 6.10. In general the plots and the D'Agostino test support the assumption of normality. Only for encoding CT:XY-SEQ is the null hypothesis that the distribution is normal rejected by the D'Agostino test. However, the plot does not show excessive variation, and due to the underlying theoretical reasoning (central limit theorem) and the relative robustness of the paired t-test, the paired t-tests with the modified variance estimator are still used.

Table 6.6.3 shows all pairwise comparisons between the predictors and their adjusted, via the Holm procedure, p-values. Of the 36 comparisons, 25 show significant differences

[10]See chapter 3

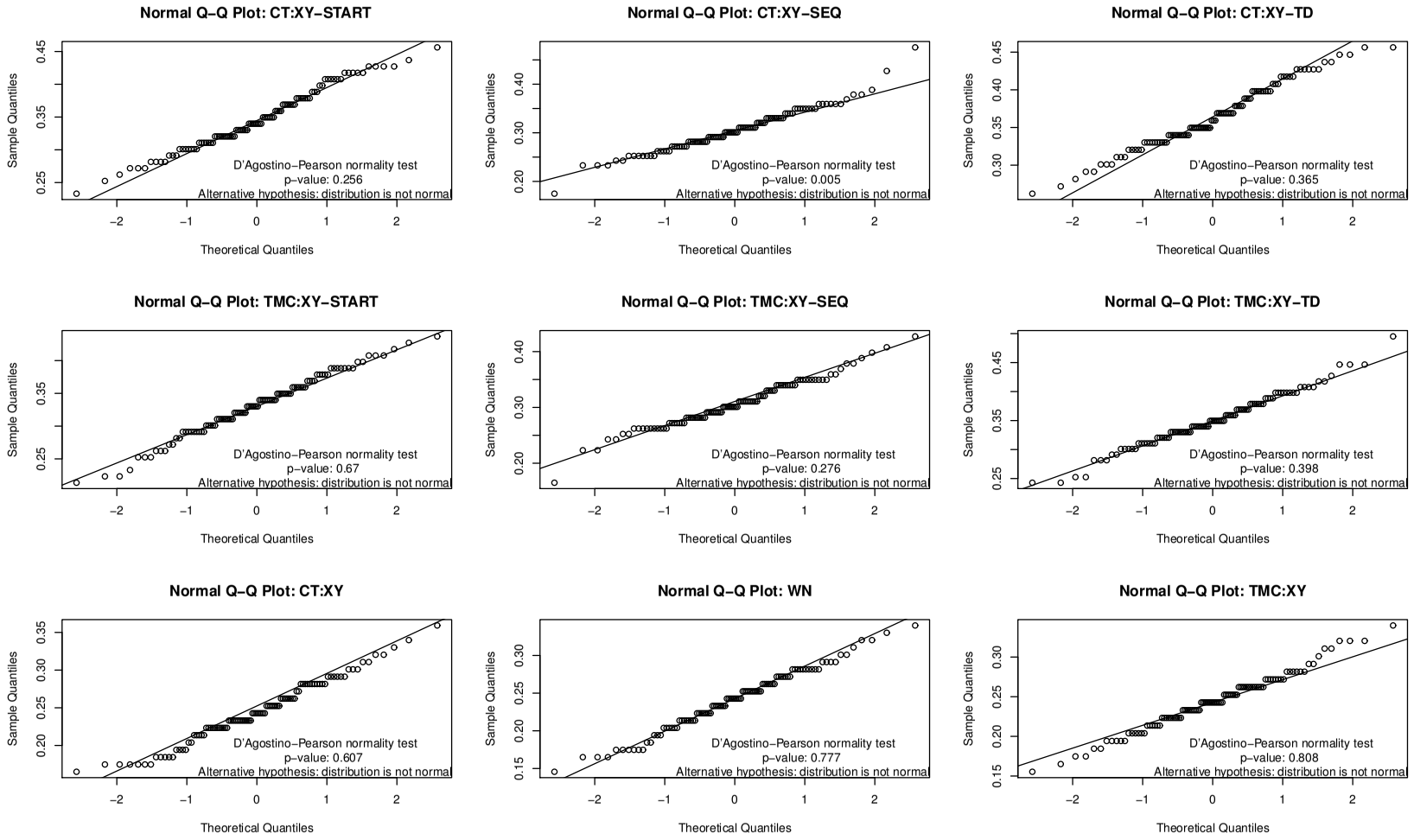


Figure 6.10: Normal probability plots and the results of the D'Agostino tests of normality.

after adjustment. Of the 25, eighteen ($p < 0.005$) confirm what was expected from the box plot, that the time relaxed encoding where no other features are used (CT:XY, TMC:XY) and the baseline WN predictors perform significantly worse than all other methods. Of these three predictors none significantly outperform any other.

Considering the global distance criteria, no significant differences were observed when only the criteria was altered. The best performing encoding was XY-TD although no significant difference was observed between it and the XY-START encoding. Compared to the encoding which assumed correct sequence information, XY-SEQ, the XY-TD encoding always performed significantly better ($p < 0.005$) as did the XY-START encoding ($p < 0.05$), except when the total minimum distance criteria was used compared to the XY-SEQ encoding also under the total minimum distance criteria.

A potential explanation for the superior performance of the total distance feature encoding is that, while deviations in accuracy regularly occurs in varying magnitudes for individual points, over the full input sequence the total magnitude may be similar. Another potential reason is that the encoding penalizes trails within the historic set which have points which are a long way from the overall path created from the trail due to the disproportional increase in total length they contribute. Since these points are generally recorded in error, historic trails with errors of these kinds are avoided as prediction candidates, potentially improving prediction performance.

Regardless of the exact reason, the result highlights (1) that the proposed approaches perform significantly better than the baseline approach, (2) that under the proposed encodings a relaxed approach to modelling sequence can be of benefit leading to improved performance, and (3) that the choice of the global distance criteria is not so important, although in conjunction with the better performing encodings it can bring performance gains. It is of note that the cover tree framework can be used to model many other feature variants with relaxed notions of sequence, some of which may perform better still.

6.7 Conclusion

In this paper a computationally efficient new approach for prediction was presented based on using multiple iterators over augmented cover trees. The approach relinquishes the use of sequence as part of the algorithm structure, in contrast to current state-of-the-

Predictors			p-value	Adjusted p-value
better		worse		
CT:XY-START	vs	WN	0	0
TMC:XY-START	vs	WN	0	0
CT:XY-SEQ	vs	WN	0	0
TMC:XY-SEQ	vs	WN	0	0
CT:XY-TD	vs	WN	0	0
TMC:XY-TD	vs	WN	0	0
CT:XY-START	vs	TMC:XY	0	0
CT:XY-TD	vs	TMC:XY	0	0
TMC:XY-TD	vs	TMC:XY	0	0
CT:XY-START	vs	CT:XY	0	0
CT:XY-TD	vs	CT:XY	0	0
TMC:XY-TD	vs	CT:XY	0	0
CT:XY-START	vs	TMC:XY-SEQ	0	0
TMC:XY-TD	vs	TMC:XY-SEQ	0	0
TMC:XY-TD	vs	CT:XY-SEQ	0	0
TMC:XY-START	vs	CT:XY	0	0.000009
TMC:XY-SEQ	vs	TMC:XY	0	0.00001
TMC:XY-START	vs	TMC:XY	0.000001	0.000011
CT:XY-START	vs	CT:XY-SEQ	0.000003	0.00006
CT:XY-TD	vs	TMC:XY-SEQ	0.000004	0.000063
TMC:XY-SEQ	vs	CT:XY	0.000012	0.000189
CT:XY-SEQ	vs	TMC:XY	0.000023	0.000352
CT:XY-SEQ	vs	CT:XY	0.000084	0.001173
CT:XY-TD	vs	CT:XY-SEQ	0.000184	0.002395
TMC:XY-START	vs	CT:XY-SEQ	0.003086	0.037035
TMC:XY-START	vs	TMC:XY-SEQ	0.015433	0.169763
TMC:XY-TD	vs	TMC:XY-START	0.02077	0.207703
CT:XY-TD	vs	CT:XY-START	0.050822	0.457396
TMC:XY-START	vs	CT:XY-TD	0.053559	0.457396
CT:XY-TD	vs	TMC:XY-TD	0.195821	1
CT:XY-START	vs	TMC:XY-START	0.200263	1
TMC:XY-TD	vs	CT:XY-START	0.285126	1
CT:XY	vs	TMC:XY	0.718813	1
CT:XY	vs	WN	0.815447	1
TMC:XY	vs	WN	0.890435	1
CT:XY-SEQ	vs	TMC:XY-SEQ	0.932019	1

Table 6.1: Table showing the pairwise comparisons of the predictors (paired t-test using the conservative OET variance estimator from chapter 3) with both the p-value and the adjusted p-value (via Holm's procedure) when the relevance parameter, α, is set to 5m. All p-values are shown to six decimal places. Better/worse column ordering is via means (see figure 6.9).

art proposals, instead allowing it to be modelled as part of the standard feature vector along with the spatial information. Allowing a wide range of previously unexplored encodings four of these were investigated under two global distance criteria and one baseline encoding, a predictor recently proposed in [137].

Between predictors the experiments showed statistically significant improvements when using encodings that provide a relaxed encoding of sequence compared to encodings that assume correct sequence information. In all cases the new encodings showed significant performance increases over the baseline, highlighting not only that the relaxed encoding is important but that the specifics in the predictor, such as keeping the matching as an independent step and the exact distance function used must be carefully considered.

Overall the results highlight the superiority of the proposed predictors in the investigated context compared to previous work and the potential of the approach to relaxed sequence encoding in general. Furthermore it must be noted that many other unexplored encoding schemes can be implemented in this framework, providing a platform for interesting future work. With respect to the computational complexity the results inherited from the use of the cover tree data structure provide solid theoretical results with the runtime complexity logarithmic in the number of unique points in all historic trails, albeit with a large constant. This logarithmic runtime performance was clearly demonstrated in practice, with significantly less distance calculations required with respect to a brute force approach, even at small numbers of training observations. This result demonstrates the practical applicability of this approach to large data sets.

Chapter 7

Click logs: Beyond GPS data

This chapter is based on work presented at the Web Science Conference in Athens [181] in 2009. As indicated by the date, the work was performed before investigation into GPS logs. While the presented results supports the use of click logs, sufficient real world data was unavailable at the time for further investigations and therefore GPS data was focused on instead. Experimental design and statistical analysis in this chapter is joint work with Dr. Chris Brien.

Prediction of movement is not only useful in the physical world. One area which has seen a significant amount of interest is the prediction of movement on the Internet. This is in part due to the proliferation of logs recording movement in this domain, with most online activity logged in some form or another. Examples include proxy or web server logs (e.g. [126]), click logs (e.g. [3]), cookies (e.g. [88]) or various other opt in (or otherwise) schemes. While in a lot of cases the recording of such data raises privacy issues (as, for example, discussed in [29, 100, 135]) the practice is wide-spread and provides a wealth of information from a data mining perspective, where in many cases privacy issues can be avoided by using sufficiently anonymized or aggregated data. Research into such anonymization techniques is ongoing see, with respect to click logs, for example [112, 143, 155]. In this chapter a specific type of logs, click-through logs from image search, that record implicit movement trails are examined.

Click-through logs are the records of the results that users have clicked on within a search engine as a result of a query. As such they represent the user's choice of navigation though the information (resource) space, labeled by the query, as provided by the search

engine. The specific path that the user takes though the information provides a level of implicit knowledge with respect to the resources. A large body of work exists with respect to utilizing this information. In the case of the similar click logs from traditional web search a primary example is the re-ranking of results. In other words to predict the next resource the user will want to move to and enable them to get their quicker by moving the result up into the top ranks (see, for example, [35, 48, 54, 82, 204]). Despite their common usage the actual navigation information provided by click logs from web document search has been shown to be very noisy [98, 171, 176] and may not encode positive information from which it is wise to base future predictions on. Unlike spatial movement, where the user can typically see, or at least has a good idea of where they are going in the majority of cases, navigation in web search results is less obvious to the average user and many false trails tend to be taken before the information they are seeking is reached. This leads to a large number of cases where the assumption that a click provides an indication of relevance of the document to the query is violated. Without taking this into account this makes document search click-through data a poor basis for prediction models. In contrast to document search, the decisions within image search can be seen to be potentially more informed. This is due to the snippet text from which the decision being made being replaced by a thumbnail of the whole image [37, 47], enabling the user to better see the resource that will be obtained by making the click and following the link, thereby reducing false trails. Conceptually similar, and holding many potential uses, image search click-through data has not seen such close examination, despite vastly different properties. Addressing this, findings from a user study involving over 67 participants are detailed. Aimed at determining the reliability of image search click-through data, factors identified in previous studies in document search click-through data are examined and results compared. In addition, additional factors uniquely present in web based image search are motivated and examined.

7.1 An Introduction to Click-through Data

The by-product of search engine interactions - the records of what users have clicked as a result of a query - known as "click-through data" is an increasingly popular resource of implicit feedback based on the general navigation of an ever increasing number of users within search engines. Recently, however, the validity of such feedback has been questioned [67, 99, 171]. Previously advocated for use in things such as: learning user

models; query suggestion; improving search; and creating query hierarchies, a number of papers have shown greater consideration needs to be taken when evaluating the validity of such feedback. However, the research has largely focused on traditional document web search. In this case users are only presented with short documents excerpts to evaluate the resource before clicking - and creating the "implicit feedback" between the query and resource. In contrast, other search types vary significantly with respect to what the user is presented with, and hence what is used to pass judgement, and create this "implicit feedback". Such variations are important as even small differences in such "caption features" can have significant impacts on user behaviour [43]. Of note is web-based image search interactions where the user is presented with a reduced thumbnail of the whole image.

Image search click-through data has also been shown to have many potential uses in recent works [37, 47, 173, 186]. Unlike traditional web search interactions, however, there has been no previous research that has explicitly looked into its validity as implicit judgements. This work presents a study of such data enabling a comparison with what has been discovered for traditional document search [67, 99, 171]. In addition searches for different types of images are evaluated. Such an evaluation is motivated by the impact of small caption features in traditional web search [43]. While conceptually similar, the results of previous studies based on web search interactions can not be directly transferred to image search interactions due to the aforementioned difference in the summary displayed and used by the user in the decision to click.

7.1.1 Related work

Implicit feedback in the form of click-through data was initially exploited in 1995 with [123] using it to help the proposed system determine relevant links for the user. Since then it has been proposed to aid a number of problems including training data for learning user models [5, 12] and document retrieval functions [97], for improving search [4, 201], for query suggestion when comparing complex question style queries [194], clustering related webpages and queries [13] and more recently in auto-generating query hierarchies [175]. However, the theoretical foundation of the use of click-through data as a source of implicit human relevance judgement has recently been questioned. In 2005, [67] examined click-through behaviour in regard to standard document web search. It was found that, used in conjunction with other implicit feedback (such as dwell time),

judgements that correlated well with explicit judgments could be determined. Used in isolation, however, such a correlation did not occur, with users satisfied with the returned document only 39% of the time after clicking a link. Also in 2005 [98] reject the use of click-through data for absolute relevance judgement, instead indicating its usefulness as relative judgements, i.e. clicked data is more relevant than un-clicked data. More recently in 2008 [176] found proposed search engine document reordering techniques based on click-through data do not always lead to improvements in search quality and may even have a detrimental effect. Also in 2008 [171] presented more evidence against the use of click-through data reporting only a 52% correlation between clicked documents and those the users thought relevant and a 58% agreement rate between the click-data and TREC judgements for the collection used in the study.

All of the above mentioned studies were conducted using click-through data from document web search interactions. However, click-through data is not solely generated from such searches, with image search being one such alternative. Such searches vary from traditional document web search in a number of critical ways. Typically image search results are a set of thumbnails, sometimes with short captions. Such summary information is intuitively more complete than the two or three line summary information displayed for traditional web search with a number of researchers agreeing with such an intuition. The primary argument put forward is that thumbnail summaries contain both more information and that the information can be absorbed faster, resulting in less poorly informed clicks than traditional web search [37, 47]. As such, results based on web document search click-through data should not be arbitrarily generalized to all forms of click-through data. Addressing this, the present work details a study focusing on image search click-through data based on similar studies into traditional web search click-through data [67, 99, 171]. The focus on image search click-though data is motivated primarily by two factors: firstly due to its provision in popular commercial search engines such as Google and hence the existence of such data; secondly due to recent work in the research community using such data.

Image search click-through data has been proposed for a number of uses. In 2005 [186] proposed the use of click-through data as a way of removing the noise introduced by polysemy. In 2006 [37] proposed the use of click-through data as relevance feedback for a technique across visual and textual features in image search. In 2007 [47] looked at a technique for addressing the sparsity problem within click-through logs. They succinctly highlight some of the major uses for click-through data, these being improving

search, query suggestion, resource annotation and relevance feedback. None of the work, however, adequately addresses the underlying question of the implicit judgments' accuracy, either in the general case or under the varying conditions associated with search results and query types. [37] motivate their choice of click-through data based on intuition alone and validate their results using artificially generated click-through data. [186] neither use real world data, nor cover many types of queries. Finally, while using significant quantities of real world data, [47] do not focus on measuring the accuracy of such feedback in general, but rather seek to improve it.

7.2 Potential factors affecting image click-through data

The presented study evaluates four factors with the potential to impact user click behaviour in web image search and hence impact the accuracy of the associated click-through data. The first is derived from observations and studies into document search based click-through data. In document based search, [99] highlighted the impact of the quality, in terms of query relevant content presented per page (system precision), of the system on the accuracy of click-through data. While the results from [171] do not seem to support such an assumption, they do not evaluate true extremes of precisions of the underlying system. Regardless such a factor has the potential to affect image click-through data and as such is included in this study.

The remaining three are motivated by the impact of "caption features" on click-through data as reported with regard to document search based click-through data [43] and the vastly different properties image queries can have [61]. Since image search results typically are returned in the form of a series of thumbnails, "caption features", as referred to in traditional document search, can be seen to be qualities of the image thumbnail. We acknowledge that in some image search systems results are returned accompanied with text snippets, however, this is not always the case. In this instance we choose to examine the systems without text. The four qualities of the image that we highlight as potential influencing factors have been motivated by research in both image and image query classification [10, 57, 91, 93, 94, 101, 174].

The first quality selected as a factor is based on categorization proposed initially by [174] and reiterated and refined more recently in [10, 93]. According to this, three categories

are identified, *general*, *specific* and *abstract*. In defining the *generic* category we adopt the definition as stated by [93]. Specifically queries falling into this category are those which only require "general everyday knowledge" to recognize the objects or scenes. In contrast, queries falling into the *specific* category refer to things that can be identified and named [93]. As such they require a greater level of knowledge (even though this knowledge might be common knowledge). An example includes the query *person* at the generic level and then *Kirsten Dunst* at the specific level. When a user searches they may or may not have this knowledge and this forms another potential factor affecting image search behaviour and hence click-through data. As such we evaluate the two extreme cases, where the user has the knowledge (known) and when the user does not (unknown). We investigate this due to the potential difference in search behaviour, and hence click-through data, that may occur as a user uses the search as a way of learning as well as acquiring images. These facets do not apply to the category of *generic* as by definition it only requires "general everyday knowledge" [93]. The third category, "About", as proposed originally by [174] is not evaluated due to its interpretative and subjective nature, which prevents accurate ground truthing. Rather, the scenarios described by researchers in this image categorization are described to a level of detail such that they then fit into one of the former two categories. The motivation behind this is two-fold. Firstly, as mentioned, objective evaluation is not possible due to the subjective nature. Secondly interpretation and subjectivity can be seen to simply lead to a lower/variable mean average system precision since, in the case of a search engine that returns only a singular and uninformed interpretation of the user's intention, the difference from the search engine and the user's subjective interpretation only reduces the perceived system precision. For example happiness, once the subjectivity is removed by a concrete description of imagery, such as "any image containing a person smiling" can then be seen to be part of the Generic Action category.

The third factor is drawn from a further breakdown of the above categories as proposed by [57, 93] who make a distinction between objects and actions and scenes. The subcategories objects and scenes are investigated as they change the visual pattern that a user is looking for when searching. The action category is not investigated since actions can, in general, be represented as a more specific object description, such as birds flying.

A final factor that, while present in the literature, we do not investigate is the distinction between visual (such as colour and texture attributes) and conceptual descriptions of images. The reason for this is the lack of prevalence as a category of actual image search

queries with [94] reporting that less than 2% of web based image queries fall into this category.

7.3 Experimental Design

Web document search click-through data has been evaluated from a number of angles using a variety of approaches. [67] developed a plugin in order to record several types of user behaviour, explicitly requesting user feedback about each selected result in two real world search engines. Based on the explicit judgements they were able to calculate the overall accuracy of the click-though data as 39%. Potential reasons for accuracy levels of click-through data, however, were not investigated. In contrast [99] specifically examine the reliability of the implicit feedback given, concluding that click-through data is affected by both the quality of the search engine and the trust the user has in the system. Their system was also backed by a commercial search engine and a proxy server used to manipulate results. Differing from that of [67] they attempted to control factors they considered to influence search behaviour, and hence user clicks. Additionally considering eye tracking data the experiment was performed in a laboratory setting. More recently [171] investigated click-through data in a more controlled fashion, generating and manipulating the results based on ground truths derived from the TREC WT10g collection. Such control was used to enforce levels of system performance in order to investigate how click behaviour varies as the quality of the underlying search system changes.

The previous experimental designs mentioned involve either setting or observing image search tasks and the experiment presented here adopts the former. Therefore a controlled search system was set up surrounding a set of fixed topics (queries) and results populated with results that were known to be either relevant or not. This resulted in a completely controlled system where the goals of evaluating differences in system performance and query types were able to be achieved.

In this work the main goals were:

1. To investigate the overall click-through accuracy of image search click-through data

2. To examine the impact of the quality of the system on click-through accuracy

under differing levels of system accuracy

3. To determine if any of the factors identified as potential influencing factors (see section 7.2) affect click-through accuracy

The first is a general goal and a similar experimental design to [171] is used making the results somewhat comparable to the 58% accuracy reported there. The second, in a similar fashion to [171], utilized the known ground truths of the images with respect to the fixed queries systematically manipulating the results to a fixed system precision for each subject. In the study two levels of system precision were investigated, one low and one high in order to examine the effect. The specific levels investigated were 16.6% and 83.3% which represents 2 out of 12 and 10 out of 12 correct images per page respectively. Each participant only evaluated the system under one precision level. The third goal was investigated by using a fixed set of queries. The additional factors identified as potential influencing factors (see section 7.2) were then encapsulated in six categories (shown in table 7.1 along with the chosen topics) generic scene, generic object, known specific scene, known specific object, unknown specific scene and unknown specific object. For each category 3 topics were selected.

Topics were based on the style of topics from ImageCLEFphoto 2006 [81] consisting of both a title and a narrative. Where possible, topics and corresponding positive images from the ImageCLEFphoto 2006 data set were used. This was only possible for the *generic* category topics due to the nature of the data set. *Specific* category topics were then developed aiming to maximize a known/unknown response in a pre-topic questionnaire, taking into account typical query types as reported by [94] mimicking the format of the topics from ImageCLEFphoto 2006. The selected topics are shown in table 7.1.

When selecting topics from the ImageCLEFphoto 2006 data set all topics with more than 100 available image ground truths were manually categorized into the six categories of interest to the study. From each category, topics were then randomly selected. Positive images for each selected topic were then randomly selected from the list of relevant images from the ground truth set from ImageCLEFphoto 2006. Positive images for topics not part of the ImageCLEFphoto data set were generated in a similar way to the relevance assessments for the ImageCLEFphoto data, by the (two) topic creators judging all images using the ternary classification scheme: (1) relevant, (2) partially relevant, and (3) not relevant. Based on these judgements, only the images judged relevant by

both assessors were considered to be relevant [46]. The pool of images evaluated by the judges in this study were based on the results from a Google image search for the given topic. For all topics negative images were identified to populate the controlled search system under the same approach, except that images were assigned to the negative set if both assessors considered the images to be not relevant. In this way negative images were representative of negative images that would be found in a real world image search engine. The complete data set used in the study consisted of 2880 images (1440 relevant and 1440 irrelevant) enabling 8 pages per search task to be displayed to the subjects.

For each topic each participant was presented with a pre-topic questionnaire to establish their familiarity with the topic in which they were asked to rank their knowledge of the topic, as well as their visual knowledge, in order to enable proper processing of the known/unknown facets being examined. A screen shot of the questionnaire is shown in figure 7.2. These topics then formed the basis of the image search task presented to the participants. Participants were given all 18 tasks. For each task they were given both the topic and the narrative and told to imagine that they were searching for images on the given topic, with the scenario given by the narrative in mind. Their goal was then to "save" up to six images for "future use" in, for example, a poster or presentation. "Saving" was achieved by clicking a button provided on the webpage containing both the full image and the website the image came from which was displayed after clicking on a result. Saving was then used to indicate that the user had considered the image relevant, enabling a user indicated relevance rating for the clicks in addition to the ground truths within the system.

Since each subject could only perform the tasks at a single level of system precision and to address possible learning effects over the tasks a randomized approach was taken with respect to participant allocation and topic completion order. This was done in groups of six, with three randomly chosen subjects getting one level and the other three the other, enabling additional blocks of subjects to be added as the participant numbers increased. The effect of this block grouping was then examined in the statistical analysis but its inclusion in the model found to be unnecessary.

In order to evaluate the impact of the potential factors discussed in section 7.2 statistical analysis was performed based on a Generalized Linear Mixed Model (GLMM) using the binomial distribution with the logit link. All analysis was performed using

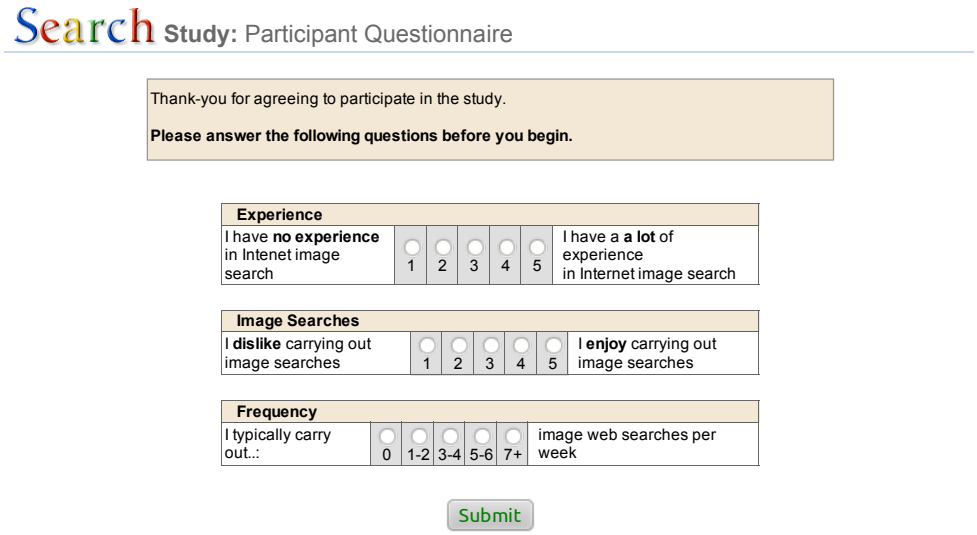


Figure 7.1: Screen shot of the initial participant questionnaire.

GenStat [2]. Initially, random terms were included to allow for random variation between the randomization groups of (six) subjects, between subjects within groups and between tests within subjects. In all cases, the between groups term was found to be unnecessary. Additionally the need for an overdispersion parameter was investigated in case the variation in the data was greater than expected for binomially distributed data. An overdispersion parameter was found to be necessary when either ground truths or user defined relevance was used to indicate click-through accuracy. The adequacy of the model was checked using simulated envelope plots and residual and absolute residual versus mean plots. Additionally the normality of the subject effects was assessed using a normal probability plot. The plots all indicated that the model was reasonable for the both the ground truth and user defined click-through accuracy variables.

The significance of model terms associated with the system precision and the different categories was assessed using approximate F tests described in [2, §5.3.6]. Differences between levels in results due to the system precision and categories were examined using predicted values from the model, and the standard errors of differences (s.e.d) were produced. A difference in the predicted mean (on the logit scale) that was more than twice its s.e.d. was regarded as being significant.

Identically to [171], but in the context of image search rather than document search, each participant was also asked to indicate their level of experience in Internet image search. Specifically participants were asked whether they liked carrying out image searches on the web and how often they typically carried such searches. A screen shot of the questionnaire is shown in figure 7.1.

Category	Topics
Generic Scene	group in front of mountain landscape tennis player on tennis court winter landscape
Generic Object	straight road bird flying photos of dark-skinned girls
Specific Scene (Known)	Paris, France London, England Sydney, Australia
Specific Object (Known)	George W. Bush Coca Cola branded can Eiffel Tower
Specific Scene (Unknown)	Baku, Azerbaijan Quito, Ecuador Tbilisi, Georgia
Specific Object (Unknown)	Shwezigon Pagoda Ali Abdullah Saleh (President of Yemen) Ushabti/Shabti/Shawabti

Table 7.1: The six categories evaluated and their corresponding topics

7.4 Results

The study ran for approximately one week as a 1 hour voluntary online study which was advertised primarily to university students and staff. As an incentive a prize draw was conducted with an entry given to participants who completed the study. The study was able to be stopped at any time and, if desired, resumed later. 116 people participated, of whom 67 completed the entire study. As in [171] a pre-experiment question was administered aimed at establishing the participant's familiarity with online image searching. Most participants considered themselves to have average experience at using image search engines, with a mean rating of 3.5 on a scale of 1 (no experience) to 5 (a large amount of experience). The mean rating for performing image searches was only 2.9 times per week. Participants reported a mean enjoyment rating of 3.4 also on a 1 (dislike) to 5 (high enjoyment) scale. Such results differ somewhat from those

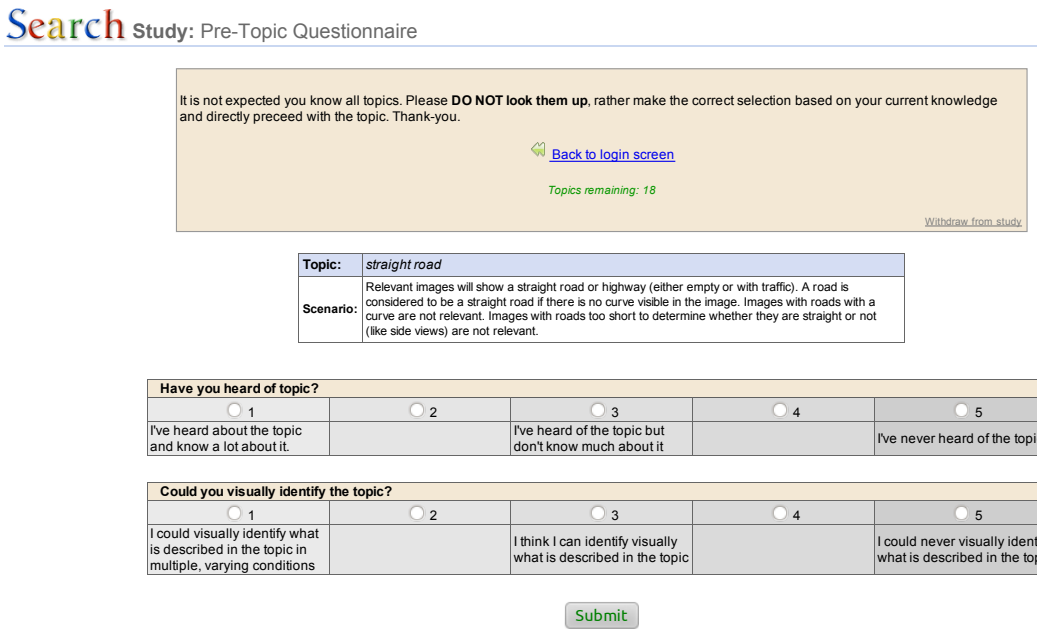


Figure 7.2: Screen shot of the pre-topic questionnaire for the topic *straight road*.

reported by the participants in [171] with regard to traditional document search who indicate mean ratings of 4.7, 7+ and 4.2 for experience, number of searches per week and enjoyment respectively on the same scales.

In the analysis only the 67 subjects who completed the entire study were considered. Additionally, based on the subjects response to the pre-topic questionnaire, subjects that indicated that they visually knew a topic that was part of a category being considered an *unknown* topic had their results for that topic moved to the corresponding *known* category (and vice versa). A participant was defined to visually know a topic if they answered either 1 or 2 to the visual knowledge question in the pre-topic questionnaire (as shown in figure 7.2) and to not visually know a topic if they answered 4 or 5. If the subject chose a 3, their response for the topic was removed from the analysis. This ensured that the factor related to the query being known or unknown was evaluated correctly.

The results of the study show an increased level of accuracy of the clickthrough data. When users indicated the relevance of the clicked image, the results show that clicked results are relevant 89.0% of the time. This is markedly higher that the 39% satisfaction with clicked document search results reported by [18] and the 52% clickthrough document relevance reported by [171]. Interestingly the accuracy found by looking at the proportion of images with relevant ground truths is slightly lower (87.6%). At the category level, the reason for this difference becomes more apparent with the results

showing the accuracy is not uniform across search categories and knowledge levels of users when system precision is low, as shown in table 7.2. Since the difference in perceived accuracy of the clicks is greater for the categories where the subjects did not think they could visually identify what was described by the topic, one possible explanation is that when the system precision was low the participants had a hard time learning the difference between object/scenes that, while similar, are strictly not correct. An alternative explanation is that participants simply clicked and marked images as relevant to complete the study. Such an explanation is unlikely for a number of reasons. Firstly to complete the study participants did not need to click any images. Rather they could simply indicate that they had finished the task. Secondly if this was the case it would be expected that the difference between participant indicated, and ground truth relevance, would be somewhat constant over all categories at low system precision which does not seem to be the case. In addition a manual examination of 20 random images selected by participants as relevant but not indicated as such by the ground truth for unknown topics lends partial evidence to former explanation, with the majority being able to be explained in this way. The presence of such results cast doubt on the reliability of subject detected relevance methods when examining the true accuracy of click-through data. In comparison when the precision of the system is high the participant-indicated relevance and ground truths are much closer (table 7.3). A possible explanation is that when users see large numbers of similar images they quickly learn the concept by placing trust in the system.

Category	P(U)	P(G)
Specific of scene (unknown)	0.8398	0.6064
Specific of object (unknown)	0.8305	0.6485
Specific of object (known)	0.8976	0.7635
Specific of scene (known)	0.8627	0.7890
Generic of scene	0.9052	0.9121
Generic of object	0.8754	0.9256

Table 7.2: Predicted means for the system precision and category interactions, transformed back to the original scale (proportions) for system precision of 16.67%. P(U): The proportion based on the number of results clicked that the subject explicitly deemed relevant ("saved") *vs.* the total number of clicks. P(G): The proportion based on the number of results clicked judged as relevant by the ground truth judges *vs.* the total number of clicks.

Category	P(U)	P(G)
Specific of object (known)	0.9068	0.9653
Specific of scene (known)	0.9195	0.9759
Specific of object (unknown)	0.9301	0.9789
Generic of object	0.8809	0.9810
Generic of scene	0.9146	0.9824
Specific of scene (unknown)	0.9216	0.9834

Table 7.3: Predicted means for the system precision and category interactions, transformed back to the original scale (proportions) for system precision of 83.33%. P(U): The proportion based on the number of results clicked that the subject explicitly deemed relevant ("saved") *vs.* the total number of clicks. P(G): The proportion based on the number of results clicked judged as relevant by the ground truth judges *vs.* the total number of clicks.

7.4.1 Statistical analysis of the factors

Relevance based on user reported relevance

In the case of relevance based on user-reported relevance[1], results showed significance in the system precision by category interaction using the approximate F statistic[2] ($p = 0.016$) and the conclusion that the difference between the two system precision levels is not the same for all categories is further investigated[3]. Upon further investigation only for the categories *specific object (unknown)* and *specific scene (unknown)* were the differences between system precision levels significant ($p < 0.05$), with category *specific scene (known)* nearly significant.

This results indicate that in general the system precision has little effect, only causing a change in click-through accuracy when the queries were unknown. A potential explanation for this was discussed previously, that a low system precision does not allow users to accurately learn when they do not know exactly what they are searching for looks like. However, questions were raised with regard to whether the user defined relevance

[1]Recall user-reported relevance was based on whether the subject chose to *save* the image or not after a click

[2]The requirement of the random Subject term in the model was checked, with it turning out to be required since the deviance decreased by 72% when it was included, compared to when it was not.

[3]It could also be concluded that the differences between categories for the low precision system are not the same as those for the high precision system

can not be relied on. While not effecting the ability to show a significant difference in the case of the *unknown* category this could potentially mask the significant differences in other categories. This is further discussed in conjunction with the results of relevance based on the ground truth relevance below.

Relevance based on ground truth relevance

In the case of relevance based on ground truth relevance results also showed significance in the system precision by category interaction using the approximate F statistic[4] ($p < 0.001$). Once again the conclusion that the difference between the two system precision levels is not the same for all categories is further investigated. Table 7.4 shows the predicted means for the different combinations of system precision and categories, transformed back to the original scale of proportions. From the table it clear to see that the differences are smallest for categories *generic object* and *generic scene* and greatest for categories *specific object (unknown)* and *specific scene (unknown)*. For all categories the differences between the precision levels in this case were significant ($p < 0.05$). This is in contrast to when considering user defined relevance where only the categories *specific object (unknown)* and *specific scene (unknown)* were significant. Considering the average precision shown in tables 7.2 and 7.3 it is of interest that in the low precision system, user relevance was higher than the ground truth but under the high precision system, the user indicated relevance precision was lower. This seems to indicate that when limited options were available the users were less fussy and were happy to subjectively accept images that may have been close but not exactly relevant to the topic. In contrast when many images were correct, they were more selective, only indicating relevance if the image met some subjective, relative criteria. An alternative explanation of course is that the ground truthing was incorrect, although given the results this would mean that the ground truths contained false positives under high system precision and the opposite, false negatives, under low system precision. Since the same images were recycled randomly in both cases this is considered unlikely. Considering this observation coupled with the previous discussion regarding the large increase in perceived relevance when the query is unknown the doubts with regard to the merits of using user defined relevance to measure click-through data are increased. Therefore further investigation with this user-defined measure of click accuracy is not considered.

[4]The requirement of the random Subject term in the model was checked, with it turning out to be required since the deviance decreased by 58% when it was included, compared to when it was not.

Category	System precision	
	16.67%	88.33%
Generic object	0.9256	0.9810
Generic scene	0.9121	0.9824
Specific object (known)	0.7635	0.9653
Specific object (unknown)	0.6485	0.9789
Specific scene (known)	0.7890	0.9759
Specific scene (unknown)	0.6064	0.9834

Table 7.4: Predicted means for the system precision and category interactions, transformed back to the original scale (proportions).

Since the interaction of system precision and categories was significant the system precision and category combinations are examined under accuracy levels defined by the ground truths. Under a system precision of 16.67% all category pairs showed a statistically significant difference except for (*generic object*, *generic scene*) and (*specific object (unknown)*, *specific scene (unknown)*). Therefore under a low system precision it can be concluded that the *generic* queries have a significantly higher click-through accuracy than all others and that searches for *specific known* queries result in higher click-through accuracy than searches for *specific unknown* queries. Differences in click-through accuracy only existed between the *scene* and *objects* categories when the queries were known. These results are in line with the general intuition with people performing known searches not having to go through trial and error clicks to learn to recognise the object or scene they are looking for. The drop in the predicted means between the generic and specific known categories is larger than expected though, since under the aforementioned explanation, in both cases people should know what they were meant to click on. With respect to the objects vs. scenes the differences seem to be less marked. While a significant difference was seen between *specific object (known)* and *specific scene (known)* where the scene category was indicated to have the higher click-through accuracy in the predicted means (see table 7.4) the difference between the predicted means is only approximately 2.55% which is relatively little compared to the differences between the other categories shown significant differences.

Under a system precision of 83.33% the story is very different with no significant difference shown. Only the pairs (*generic scene*, *specific object (known)*) and (*generic object*, *specific object (known)*) can be considered to be close to showing significant differences,

with the *specific object (known)* having the lower predicted click-through average accuracy in both cases. Considering the predicted means (table 7.4) this is due to the high performance under all factors. This indicates that under high system precision the factors do not affect the click-through relevance. This is as expected since when system precision is high, even if the topic is unknown, and assuming users do learn, the first few random clicks are more likely to return positive results from which this learning can occur.

Comparing the results under both system precisions it is clear that the factors generic versus specific and known versus unknown have an effect at low system precision levels but the effects are not significant when the system precision is high. Considering this and the possible reasons for the differences at the low levels of precision previously discussed it is highly likely that these differences will become gradually less pronounced as the system precision increases.

7.5 Conclusions

Click-through data represents a record of choices within the navigation of search results on the Internet. Initially assumed to provide relevance judgement their use has been widespread with numerous applications and techniques proposed aimed at utilizing the logs of previous people's decisions to predict future decisions. The assumption of clicks providing relevance judgements has been addressed in traditional web search showing poor accuracy. However, the utility of image search click-through data has not previously been investigated. To this end this chapter provided evidence that, while also suffering from quality bias (varying accuracy under differing levels of system precision), the average precision levels are significantly higher. Specifically the reliability of web image search click-through data was investigated in a comparable way to evaluations in traditional web-based document search via an in-depth user study. Based on similar studies into document search click-throughs by [67, 99, 171] factors were examined which were shown to undermine the reliability of such data as well as additional factors uniquely present in image search. The results of this study indicate that image search click-through data is considerably more accurate in general than document based search click-through data (87.6% vs. 58% on average). Additionally the study showed that, at low levels of system precision, factors of knowledge (known or unknown queries) and

whether the queries are specific or generic systematically affect the resulting level of relevance in the click data. In general, generic queries resulted in higher levels of accuracy while unknown queries performed the worst. If the latter category is excluded then the average accuracy of the click data is predicted to be much higher, with even the worst case set of factors and system precision (low system precision, known specific object) showing 76.3% accuracy. This is significantly higher than the overall accuacy of 58% of traditional document search.

Chapter 8

Conclusion and Future Work

8.1 Conclusion and Contributions

Recent advances and the widespread uptake of location-aware consumer devices has led a growth of research into movement prediction based on digital records of movement. Specifically this thesis considered the prediction of fine-grained route prediction at the level of pedestrians. In contrast prior work has mostly focused on coarse-grained predictions [38, 127, 146, 148, 182], predictions of high-level movement from predetermined or mined regions of interest [137, 147], predictions from low sampling rates (e.g. 20 minutes [177]), vehicle predictions [69, 116, 200] or only on predicting the next location or destination [31, 55, 182, 200]. Predicting a detailed route for pedestrians is an important goal. Over and above destination prediction it provides additional invaluable information that can be used by services and potentially intelligent mobile agents to not only display or provide important information to the user ahead of time, but also to plan and prepare information in advance and in the most resource-efficient way possible to ensure seamless operation in the presence of connectivity, power and other constraints inherent in the mobile environment.

To this end this thesis extended state-of-the-art in pattern matching based approaches to fine-grained pedestrian route prediction based on mining GPS data. Laying the foundation, previously proposed statistical prediction methods proposed under similar contexts were discussed with a focus on route prediction from GPS data in chapter 2. Following this the thesis has provided the following main contributions:

1. A set of current *best practice* procedures for the task of statistical analysis of

movement prediction results, including the recommendation and motivation for an alternative distance measure (Chapter 3).

2. A computationally efficient algorithm for predicting pedestrian routes from GPS data aimed at resource-limited mobile devices which showed performances close to much more complex, full order predictors in experiments under 5m grid quantization (Chapter 4).

3. An investigation into the use of measures other than conditional probability for measuring the utility of predictions in large data sets where the input matches multiple different historic trails equally in light of theory from the data mining community (Chapter 5).

4. Positive results and a new prediction model and set of feature encodings questioning the intuitive and state-of-the-art approaches to prediction that assume accurate sequencing information (Chapter 6).

5. The examination of the usefulness of an alternate source of incidental movement information, click-through data from search engines, representing movement within the information space delivered by search engines (Chapter 7).

These contributions are briefly summarized in the following sections.

8.1.1 Evaluation Methodology: Best practices

In chapter 3 the challenges of evaluating movement prediction models are discussed. The main hurdles are the choice of the evaluation measure, for which many candidates exist, and the complexities of statistical analysis in the case where small data sets necessitate data reuse preventing normal statistical analysis. With respect to the choice of the evaluation measure, this thesis shows that measures common to the seemingly similar fields of time series prediction such as mean average precision and root mean square error are flawed in the context of movement prediction, and while a significant improvement, the metric proposed by [69] can be improved. To this end the average Fréchet distance is proposed and an implementation provided in appendix A for the **R** statistical computing software [158] based on code from [72]. With respect to the issues of statistical evaluation, it is of note that statistical analysis is not seen in any of the directly related literature. An important part of scientific comparative analysis, particularly as the number of proposed procedures increases, this thesis examines the issues

in context including some empirical evidence based on the evaluation of movement predictors in other chapters in this thesis. The chapter concludes by contributing a current best practice guide to a robust evaluation methodology with respect to evaluating route predictions.

8.1.2 A novel approach to prediction under minimal resources

Pedestrian-level movement prediction provides the potential to provide invaluable information to mobile devices operating to assist the user. However, such devices typically have very limited resources. Noting a significant gap in performance between the computationally cheap low order Markov Models and the state-of-the-art full order models that come at a significantly higher computational cost, a novel approach to prediction based on approximating a full order Markov predictor is proposed and evaluated. Utilizing a relaxed encoding with respect to sequence, under similar motivations as the investigations in chapter 6, empirical results show performance close to that of a full order model which additionally utilized a Euclidean distance matching function. Importantly the proposed predictor has space and runtime complexities in the same order, and only slightly elevated in practice, as the widely used simple first order Markov model. Specifically the more complex models have space complexities in the order of the number unique sequence orderings of all trails in the historic data set. In contrast the proposed approach has an upper bound on space complexity in the square of the number of spatial locations. State-of-the-art approaches additionally have much higher runtime complexities compared to the proposed approach which has a runtime in the order of the length of the prediction.

8.1.3 Investigations into mining movement pattens: beyond conditional probability

Movement prediction models always involve some metric to calculate how often certain patterns appeared in the historic data set. This metric is referred to as an objective value function in this thesis. In contrast to the matching mechanisms, where many different approaches have been investigated, proposed movement prediction algorithms have exclusively used conditional probability, or in one case frequency, with limited or no discussion. This is despite a large body of work from the data mining community

identifying over 30 different objective value functions [71]. In chapter 5 a comprehensive investigation is performed into the use of different objective value functions. The different objective value functions are evaluated using a proposed predictor developed to (1) address the issues of noise, (2) enabling the direct application of a wide range of metric as developed in the literature and (3) allow a wide range of matching function to be used in order to evaluate the effect of different matching functions in combination with the different metrics. After motivating the choice of various objective value functions and matching strategies 30 different variants consisting of five different objective value functions, four different matching functions and two methods of combining the two were evaluated. Primary outcomes of the experiment were results showing that rules close to an input should be prioritised over those that are further away, even when the distant rules are more highly rated with respect to their objective value. Additionally, the results showed that when this advice is taken, the choice between the objective value functions (of the type motivated as applicable) is of limited concern and conditional probability does represent a good choice. However, under certain data sets with specific properties, predictors may benefit from other objective functions such as RLD which seek to address cases where conditional probability is known to give unintuitive results.

8.1.4 Investigations into novel encodings and a corresponding predictor

How prediction models perform is intimately tied to the features used to make the prediction. Throughout chapter 4 and in 6 the assumption and use of algorithms relying on the presence of accurate sequencing information is questioned and relaxed encodings considered. This is more fully examined in chapter 6 where the computational and logical reasons are discussed, noting that state-of-the-art algorithms rely on accurate sequencing information to reduce computational complexity that would otherwise be prohibitive. Therefore in order to investigate the relaxed encoding of sequence a new framework for prediction is developed and presented based on multiple iterators on an augmented version of a data structure called cover trees [21]. The approach allows the definition of sequence and time as a feature at the same level as any other, such as location. Using the data structure and an alternative global distance function, a number of time/sequence relaxed encodings are considered. Computational efficiency and superior prediction accuracy is then shown compared to sequential encodings used

by state-of-the-art predictors and the predictor proposed in 2009 in [137]. It is of note that the encodings evaluated are only a few of the many possible with others easily implemented.

8.1.5 Beyond GPS data: an evaluation of other data sources

Movement does not only occur in the physical world. One area which has seen a significant amount of interest is prediction of movement in the World Wide Web, partly due to the large proliferation of logs. Differing in many ways from movement data from the real world the data sources poses additional challenges. In chapter 7 click-through data from image search engines is considered. The click logs in this context represent movement though the information space presented by the search engine. However, in the case of web document search click-through data, the clicks have been shown to encode many poor choices of movement, i.e. that there is limited intelligence being contributed from each individual, with the clicks only leading to documents relevant to the query in 58% or less of the cases. Based on the theoretical justification that image search click-through data should be more reliable due to better caption information, the thumbnail from which the user typically makes their click decisions, an in-depth user study involving over 67 participants was run, additionally investigating a number of other factors with the potential to affect click accuracy. The result showed that image search click-though data is indeed more reliable than document search click-through data although it is still susceptible to a number factors, such as the precision of the search engine.

8.2 Future work

In contributing two new prediction algorithms, a set of best practice evaluation guidelines, an investigation into alternatives to conditional probability and an investigation into the use of click-through data numerous avenues for additional work have been exposed. In many cases these are noted within the chapters themselves. These are reiterated here along with additional interesting points of future investigation.

8.2.1 Improvements to the evaluation methodology

Due to the nature of systematic data reuse, a conservative approach to estimating the variance and hence conservative inference was recommended. This is important in order to prevent researchers reporting that predictors are significantly better than others when in fact they are not (inflated Type I error). This is at the cost of Type II errors where no difference is observed by the statistical procedure when one actually exists. While this is preferable than the alternative it is certainly not optimal. As noted in chapter 3 [131] provided an approach shown to be less conservative that those recommended in this thesis from [141] which is only valid under continuous loss functions which is not the case with the evaluation metric recommended for route prediction. However, the authors did show some preliminary results based on using an approximation function for non-continuous loss functions in the case of classification. The authors note, however, that further research is required before its use. Since this has potential benefit to the evaluation of route prediction techniques such an investigation, particularly with a focus on the application of movement prediction, would make both interesting and highly useful future work.

8.2.2 Additional modelling investigations

As noted in chapter 4 the effect of resampling the raw input GPS data into a consistent sample rate was only briefly considered in a small pilot study which indicated reduced performance. A potential reason is that as the path between correct and erroneous points is resampled a large number of additional erroneous points is introduced when resampling is performed at small intervals. However, a number of algorithms could potentially benefit from a fixed sampling rate which can not be guaranteed without resampling due to the nature of the GPS data. Therefore a full formal evaluation is marked for future work.

8.2.3 Utilization of the batch query algorithm proposed for Cover trees

In chapter 6 where augmented cover trees were used to perform efficient matching between unordered points multiple iterators were used over the cover tree structure. In

the original proposal of the cover tree [20] a batch query algorithms was proposed in addition to the single query algorithm, showing a reduction in the theoretical complexity of finding nearest neighbours simultaneously. Since the multiple iterators are essentially multiple queries it should be possible to modify the batch query algorithm to take advantage of the reduced theoretical complexity. However, as noted in the chapter, this may not result in an increased performance in the case of the small number of queries typically issued at once for predictions. However, when long inputs (contexts) are used this may have a significant impact and therefore is potentially future work of note.

8.2.4 Additional sequence/time relaxed encodings

In this work only four different encodings were examined where time was considered to be an equal feature along with location. Many other encodings are possible and their investigation makes interesting future work. More broadly the investigation of encodings involving many different factors and their effect on prediction are of great interest as they have to potential to make both a large positive and negative impact on prediction accuracies.

8.2.5 Evaluation of the algorithms over different data sets

It is well known that machine learning algorithms that perform well on one data set can often perform poorly on other data sets and vice versa. Therefore it is important to evaluate algorithms over many different data sets. Unfortunately, currently, limited data sets exist containing pedestrian GPS traces of sufficient size and density to reflect real world situations where such algorithms may be employed. Future work is proposed to collect additional real world data and further evaluate the algorithms. Ideally if multiple data sets became available then statistical analysis with regard to the generalization error across data sets could be performed (such as detailed in [50]) when comparing multiple predictors.

8.2.6 Application of prediction techniques beyond GPS data

In chapter 7 an alternative form of navigation data embedded in logs capturing human intelligence was examined. Showing promise, the use of statistical prediction models

in this context is open for future work, with many of the statistical models applied to document search directly relevant to the applications such as image re-rankings, although many differences (for example the layout) make different assumptions in the models required.

8.2.7 Evaluation of the effectiveness in real world applications

A final, broad category for future work is the evaluation of the effectiveness of the predictors in real world applications. Is current state-of-the-art useful? What is the minimum level that needs to be achieved? Does the use of human-orientated output, such as the heatmaps proposed in chapter 5 convey the level of uncertainty and possible alternatives in such a fashion that even when predictions are poor they are more useful than not? While a broad set of problems, they particularly interesting as advances in prediction algorithms and feature selection continue to increase the level of accuracy the systems are able to ascertain.

8.3 Final remarks

The widespread uptake and continual development of more precise mobile locational technology has led to the availability of a vast amount of mobility data which inherently contains intelligent decisions made by large numbers of people. This data source provides a rich and exciting basis from which movement prediction techniques can be developed. Considering the case of pedestrian route prediction this thesis extended state-of-the-art in prediction models and evaluation techniques, providing in-depth experimental results and a comprehensive discussion into the specific case of fine-grained route prediction over and above traditional region-of-interest, next step or destination prediction. While access to public data sets of such data is currently limited, the large amounts of data being generated every day makes the area an exciting one. Finally this list of applications of such work is extensive with a vast number of applications already proposed to take advantage of the data and the resulting predictions it can enable. These include, but are certainly not limited to, enabling intelligent mobile agents, informing software for ad-hoc networking and improving the efficiency of hybrid cars.

Bibliography

[1] *Introduction to statistical relational learning*, chapter An Introduction to Conditional Random Fields for Relational Learning. MIT Press, 2007.

[2] *The Guide to GenStat Release 11, Part 2 Statistics*, 2008.

[3] *WSCD '09: Proc. the 2009 workshop on Web Search Click Data*. ACM, 2009.

[4] E. Agichtein, E. Brill, and S. Dumais. Improving web search ranking by incorporating user behavior information. In *Proc. Int'l ACM SIGIR Conf. Research and Development in Information Retrieval (SIGIR 06)*, pages 19–26. ACM, 2006.

[5] E. Agichtein, E. Brill, S. Dumais, and R. Ragno. Learning user interaction models for predicting web search result preferences. In *Proc. Int'l ACM SIGIR Conf. Research and Development in Information Retrieval (SIGIR 06)*, pages 3–10. ACM, 2006.

[6] R. Agrawal and R. Srikant. Fast algorithms for mining association rules in large databases. In J. Bocca, M. Jarke, and C. Zaniolo, editors, *Proc. Int'l Conf. Very Large Data Bases (VLDB 94)*, pages 487–499. Morgan Kaufmann, 1994.

[7] H. Alt and M. Godau. Computing the Fréchet distance between two polygonal curves. *Computational Geometry and Applications*, 5(1):75–91, 1995.

[8] S. Altschul, T. Madden, A. Schffer, J. Zhang, Z. Zhang, W. Miller, and D. Lipman. Gapped BLAST and PSI-BLAST: a new generation of protein database search programs. *Nucleic Acids Research*, 25(17):3389–3402, 1997.

[9] L. Arge, M. Berg, H. Haverkort, and K. Yi. The priority R-tree: A practically efficient and worst-case optimal R-tree. *ACM Transactions on Algorithms*, 4:9:1–9:30, March 2008.

[10] L. Armitage and P. Enser. Analysis of user need in image archives. *Journal of Information Science*, 23:287–299, Aug 1997.

[11] D. Ashbrook and T. Starner. Using GPS to learn significant locations and predict movement across multiple users. *Personal Ubiquitous Computing*, 7(5):275–286, 2003.

[12] R. Baeza-Yates, C. Hurtado, M. Mendoza, and G. Dupret. Modeling user search behavior. In *Web Congress, 2005. LA-WEB 2005.*, page 10. IEEE, 2005.

[13] D. Beeferman and A. Berger. Agglomerative clustering of a search engine query log. In *Proc. ACM SIGKDD Int'l Conf. Knowledge Discovery and Data Mining (KDD 00)*, pages 407–416. ACM, 2000.

[14] R. Begleiter, R. El-Yaniv, and G. Yona. On prediction using variable order Markov models. *Journal of Artificial Intelligence Research*, 22(1):385–421, 2004.

[15] J. Beis and D. Lowe. Shape indexing using approximate nearest-neighbour search in high-dimensional spaces. In *Proc. IEEE Conf. Computer Vision and Pattern Recognition (CVPR 97)*, pages 1000 –1006, 1997.

[16] Y. Bengio and Y. Grandvalet. No unbiased estimator of the variance of k-fold cross-validation. *Journal of Machine Learning Research*, 5:1089–1105, Dec 2004.

[17] Y. Bengio and C. Nadeau. Inference for the generalization error. CIRANO Working Papers 99s-25, CIRANO, Jul 1999.

[18] C. Bettini, S. Jajodia, and P. Samarati. *Privacy in location-based applications: research issues and emerging trends*. Springer-Verlag, NY, 2009.

[19] K. Beyer, J. Goldstein, R. Ramakrishnan, and U. Shaft. When is nearest neighbor meaningful? In C. Beeri and P. Buneman, editors, *Database Theory ICDT99*, volume 1540 of *Lecture Notes in Computer Science*, pages 217–235. Springer Berlin, 1999.

[20] A. Beygelzimer, S. Kakade, and J. Langford. Cover trees for nearest neighbor. Extended version of *Cover trees for nearest neighbor, ICML '06*. Avaliable from: `http://hunch.net/~jl/projects/cover_tree/paper/paper.ps`.

[21] A. Beygelzimer, S. Kakade, and J. Langford. Cover trees for nearest neighbor. In *Proc. Int'l Conf. Machine learning (ICML 06)*, pages 97–104. ACM, 2006.

[22] A. Bhattacharya and S. Das. Lezi-update: an information-theoretic approach to track mobile users in pcs networks. In *Proc. ACM/IEEE Int'l Conf. Mobile Computing and Networking (MobiCom 99)*, pages 1–12. ACM, 1999.

[23] G. Bishop and G. Welch. An introduction to the kalman filter. In *Proc. ACM SIGGRAPH Computer Graphics, course (SIGGRAPH 01)*. ACM, 2001.

[24] O. Boiman, E. Shechtman, and M. Irani. In defense of nearest-neighbor based image classification. In *Proc. IEEE Conf. Computer Vision and Pattern Recognition (CVPR 08)*, pages 1 –8, 2008.

[25] S. Borra and A. Ciaccio. Measuring the prediction error. a comparison of cross-validation, bootstrap and covariance penalty methods. *Computational Statistics and Data Analysis*, 54(12):2976 – 2989, 2010.

[26] R. Bouckaert. Choosing between two learning algorithms based on calibrated tests. In *Proc. Int'l Conf. Machine Learning (ICML 03)*, volume 20, page 51, 2003.

[27] U. Braga-Neto and E. Dougherty. Is cross-validation valid for small-sample microarray classification? *Bioinformatics*, 20(3):374–380, 2004.

[28] S. Brakatsoulas, D. Pfoser, R. Salas, and C. Wenk. On map-matching vehicle tracking data. In *Proc. Int'l Conf. Very Large Data Bases (VLDB 05)*, pages 853–864. VLDB Endowment, 2005.

[29] A. Broder. Data mining the internet and privacy. In B. Masand and M. Spiliopoulou, editors, *Web Usage Analysis and User Profiling*, volume 1836 of *Lecture Notes in Computer Science*, pages 56–73. Springer Berlin, 2000.

[30] P. Buhlmann and A. Wyner. Variable length markov chains. *The Annals of Statistics*, 27(2):480–513, 1999.

[31] I. Burbey and T. Martin. Predicting future locations using prediction-by-partial-match. In *Proc. ACM Int'l Workshop Mobile Entity Localization and Tracking in GPS-less Environments (MELT 08)*, pages 1–6. ACM, 2008.

[32] P. Burman. A comparative study of ordinary cross-validation, v-fold cross-validation and the repeated learning-testing methods. *Biometrika*, 76(3):503–514, 1989.

[33] S. Bush and N. Smith. The limits of motion prediction support for ad hoc wireless network performance. In L. Yang, H. Arabnia, and L. Wang, editors, *Proc. Int'l Conf. Wireless Networks*, pages 495–501. CSREA, 2005.

[34] J. Chambers, W. Cleveland, B. Kleiner, and P. Tukey. *Graphical methods for data analysis*. Wadsworth, Belmont, 1983.

[35] O. Chapelle and Y. Zhang. A dynamic bayesian network click model for web search ranking. In *Proc. Int'l Conf. World Wide Web (WWW 09)*, pages 1–10, 2009.

[36] S. Chawathe. Segment-based map matching. In *Proc. IEEE Intelligent Vehicles Symp.*, pages 1190 –1197, 2007.

[37] E. Cheng, F. Jing, L. Zhang, and H. Jin. Scalable relevance feedback using click-through data for web image retrieval. In *Proc. ACM Int'l Conf. Multimedia (MM 06)*, pages 173–176. ACM, 2006.

[38] M. Chiu and M. Bassiouni. Predictive schemes for handoff prioritization in cellular networks based on mobile positioning. *Selected Areas in Communications*, 18(3):510–522, 2000.

[39] P. Ciaccia and M. Patella. The M^2-tree: Processing complex multi-feature queries with just one index. In *Proc. DELOS Workshop: Information Seeking, Searching and Querying in Digital Libraries*, volume 52, page 54, 2000.

[40] P. Ciaccia, M. Patella, and P. Zezula. M-tree: An efficient access method for similarity search in metric spaces. In *Proc. Int'l Conf. Very Large Data Bases (VLDB 97)*, pages 426–435. Morgan Kaufmann, 1997.

[41] P. Ciaccia, M. Patella, and P. Zezula. Processing complex similarity queries with distance-based access methods. In H. Schek, G. Alonso, F. Saltor, and I. Ramos, editors, *Advances in Database Technology (EDBT 98)*, volume 1377 of *Lecture Notes in Computer Science*, pages 9–23. Springer Berlin, 1998.

[42] P. Clark and R. Boswell. Rule induction with CN2: Some recent improvements. In Y. Kodratoff, editor, *Machine Learning - EWSL-91*, volume 482 of *Lecture Notes in Computer Science*, pages 151–163. Springer Berlin, 1991.

[43] C. Clarke, E. Agichtein, S. Dumais, and R. White. The influence of caption features on clickthrough patterns in web search. In *Proc. Int'l ACM SIGIR Conf. Research and Development in Information Retrieval (SIGIR 07)*, pages 135–142. ACM, 2007.

[44] J. Cleary, W. Teahan, and I. Witten. Unbounded length contexts for PPM. In *Proc. Conf. Data Compression (DCC 95)*, pages 52 –61, 1995.

[45] J. Cleary and I. Witten. Data compression using adaptive coding and partial string matching. *IEEE Transactions on Communications*, 32(4):396–402, Apr 1984.

[46] P. Clough, M. Grubinger, T. Deselaers, A. Hanbury, and H. Muller. Overview of the imageclef 2006 photographic retrieval and object annotation tasks. In C. Peters, P. Clough, F. Gey, J. Karlgren, B. Magnini, D. Oard, M. de Rijke, and M. Stempfhuber, editors, *Evaluation of Multilingual and Multi-modal Information Retrieval*, volume 4730 of *Lecture Notes in Computer Science*, pages 579–594. Springer Berlin, 2007.

[47] N. Craswell and M. Szummer. Random walks on the click graph. In *Proc. Int'l ACM SIGIR Conf. Research and Development in Information Retrieval (SIGIR 07)*, pages 239–246. ACM, 2007.

[48] N. Craswell, O. Zoeter, M. Taylor, and B. Ramsey. An experimental comparison of click position-bias models. In *Proc. Int'l Conf. Web search and Web Data Mining (WSDM 08)*, pages 87–94, 2008.

[49] R. D'Agostino, A. Belanger, and R. D'Agostino Jr. A suggestion for using powerful and informative tests of normality. *The American Statistician*, 44(4):316–321, 1990.

[50] J. Demšar. Statistical comparisons of classifiers over multiple data sets. *Journal of Machine Learning Research*, 7:1–30, Dec 2006.

[51] J. Diatta, H. Ralambondrainy, and A. Totohasina. Towards a unifying probabilistic implicative normalized quality measure for association rules. In F. Guillet and H. Hamilton, editors, *Quality Measures in Data Mining*, volume 43 of *Studies in Computational Intelligence*, pages 237–250. Springer, 2007.

[52] N. Dimokas, D. Katsaros, P. Bozanis, Y. Manolopoulos, and A. Zomaya. Predictive location tracking in cellular and in ad hoc wireless networks. In L. Yang, A. Waluyo, J. Ma, L. Tan, and B. Srinivasan, editors, *Mobile Intelligence*, Wiley Series on Parallel and Distributed Computing. John Wiley & Sons, Inc., 2010.

[53] D. Duarte, A. Galves, and N. Garcia. Markov approximation and consistent estimation of unbounded probabilistic suffix trees. *Bulletin of the Brazilian Mathematical Society*, 37:581–592, 2006.

[54] G. Dupret and B. Piwowarski. A user browsing model to predict search engine click data from past observations. In *Proc. Int'l ACM SIGIR Conf. Research and Development in Information Retrieval (SIGIR 08)*, pages 331–338, 2008.

[55] N. Eagle, A. Clauset, and J. Quinn. Location segmentation, inference and prediction for anticipatory computing. In *Proc. AAAI Spring Symp. Technosocial Predictive Analytics*. AAAI, 2009.

[56] N. Eagle and A. Pentland. Eigenbehaviors: identifying structure in routine. *Behavioral Ecology and Sociobiology*, 63:1057–1066, 2009.

[57] J. Eakins. Techniques for image retrieval. *Library & Information Briefings*, (85):1–15, 1998.

[58] B. Efron and R. Tibshirani. *An Introduction to the Bootstrap*. Chapman & Hall/CRC Press LLC, 1993 (reprint 1998).

[59] B. Efron and R. Tibshirani. Improvements on cross-validation: The .632+ bootstrap method. *Journal of the American Statistical Association*, 92(438):548–560, 1997.

[60] T. Eiter and H. Mannila. Computing discrete fréchet distance. Technical report, Technische Universitat Wien, 1994.

[61] P. Enser, C. Sandom, J. Hare, and P. Lewis. Facing the reality of semantic image retrieval. *Journal of Documentation*, 63:465–481, 2007.

[62] R. Fagin. Combining fuzzy information from multiple systems (extended abstract). In *Proc. ACM Symp. on Principles of Database Systems (PODS 96)*, pages 216–226. ACM, 1996.

[63] R. Fagin, A. Lotem, and M. Naor. Optimal aggregation algorithms for middleware. *Journal of Computer and System Sciences*, 66(4):614 – 656, 2003.

[64] T. Fawcett. An introduction to ROC analysis. *Pattern recognition letters*, 27(8):861–874, 2006.

[65] M. Feder, N. Merhav, and M. Gutman. Universal prediction of individual sequences. *IEEE Transactions on Information Theory*, 38(4):1258 –1270, Jul 1992.

[66] F. Ferrari and A. Wyner. Estimation of general stationary processes by variable length markov chains. *Scandinavian Journal of Statistics*, 30(3):459–480, 2003.

[67] S. Fox, K. Karnawat, M. Mydland, S. Dumais, and T. White. Evaluating implicit measures to improve web search. *ACM Transactions on Information Systems*, 23:147–168, 2005.

[68] J. Friedman, J. Bentley, and R. Finkel. An algorithm for finding best matches in logarithmic expected time. *ACM Transactions on Mathematical Software*, 3:209–226, Sept 1977.

[69] J. Froehlich and J. Krumm. Route prediction from trip observations. In *Intelligent Vehicle Initiative (IVI) Technology Controls and Navigation Systems*, volume 2193 of *Special Publications from the SAE World Congress*, pages 53–66. SAE, Apr 2008.

[70] S. Garcıa and F. Herrera. An Extension on "Statistical Comparisons of Classifiers over Multiple Data Sets" for all Pairwise Comparisons. *Journal of Machine Learning Research*, 9:2677–2694, 2008.

[71] L. Geng and H. Hamilton. Interestingness measures for data mining: A survey. *ACM Computer Surveys*, 38(9), 2006.

[72] C. Genolini. *longitudinalData: Longitudinal Data*, 2010. R package version 0.6.5.

[73] Z. Ghahramani. Learning dynamic bayesian networks. In C. Giles and M. Gori, editors, *Adaptive Processing of Sequences and Data Structures*, volume 1387 of *Lecture Notes in Computer Science*, pages 168–197. Springer Berlin, 1998.

[74] F. Giannotti, M. Nanni, and D. Pedreschi. Efficient mining of temporally annotated sequences. In J. Ghosh, D. Lambert, D. Skillicorn, and J. Srivastava, editors, *Proc. SIAM Int'l Conf. Data Mining*, number 32. SIAM, 2006.

[75] F. Giannotti, M. Nanni, D. Pedreschi, and F. Pinelli. Mining sequences with temporal annotations. In H. Haddad, editor, *ACM Symp. on Applied computing*, pages 593–597. ACM, 2006.

[76] F. Giannotti, M. Nanni, F. Pinelli, and D. Pedreschi. Trajectory pattern mining. In *Proc. ACM SIGKDD Int'l Conf. Knowledge Discovery and Data Mining (KDD 07)*, pages 330–339. ACM, 2007.

[77] M. González, C. Hidalgo, and A. Barabási. Understanding individual human mobility patterns. *Nature*, 453(7196):779–782, 2008.

[78] K. Gopalratnam and D. Cook. Active LeZi: An incremental parsing algorithm for sequential prediction. In *Proc. Florida Artificial Intelligence Research Symp.*, pages 38–42, 2003.

[79] C. Grana, D. Borghesani, and R. Cucchiara. Video shots comparison using the

mallows distance. In *Int'l Workshop Database and Expert Systems Applications (DEXA 07)*, pages 49 –53, 2007.

[80] A. Gray and A. Moore. N-body problems in statistical learning. In T. Leen and T. Dietterich, editors, *Proc. Advances in Neural Information Processing Systems (NIPS 01)*. MIT Press, 2001.

[81] M. Grubinger, P. Clough, H. Muller, and T. Deselaers. The iapr benchmark: A new evaluation resource for visual information systems. In *Int'l Conf. Language Resources and Evaluation*, 2006.

[82] F. Guo, C. Liu, A. Kannan, T. Minka, M. Taylor, Y. Wang, and C. Faloutsos. Click chain model in web search. In *Proc. Int'l Conf. World Wide Web (WWW 09)*, pages 11–20, 2009.

[83] H. Guo and W. Hsu. A survey of algorithms for real-time Bayesian network inference. In *Joint Workshop Real-Time Decision Support and Diagnosis*. AAAI, 2002.

[84] A. Guttman. R-trees: a dynamic index structure for spatial searching. In *Proc. ACM Int'l Conf. Management of Data (SIGMOD 84)*, volume 14, pages 47–57. ACM, 1984.

[85] M. Hahsler, B. Gruen, and K. Hornik. arules – a computational environment for mining association rules and frequent item sets. *Journal of Statistical Software*, 14(15):1–25, 2005.

[86] J. Hanley, A. Negassa, M. Edwardes, and J. Forrester. Statistical Analysis of Correlated Data Using Generalized Estimating Equations: An Orientation. *American Journal of Epidemiology*, 157(4):364–375, 2003.

[87] G. Hjaltason and H. Samet. Incremental distance join algorithms for spatial databases. In *Proc. ACM Int'l Conf. Management of Data (SIGMOD 98)*, pages 237–248. ACM, 1998.

[88] S. Ho. An exploratory study of using a user remote tracker to examine web users' personality traits. In *Proc. Int'l Conf. Electronic Commerce (ICEC 05)*, pages 659–665. ACM, 2005.

[89] B. Hofmann-Wellenhof, H. Lichtenegger, and J. Collins. *GPS Theory and Practive*. Springer-Verlag, Vienna, fifth, revised edition edition, 2001.

[90] B. Hoh, M. Gruteser, H. Xiong, and A. Alrabady. Preserving privacy in GPS traces via uncertainty-aware path cloaking. In *Proc. ACM Conf. Computer and Communications Security (CCS 07)*, pages 161–171. ACM, 2007.

[91] L. Hollink, A. Schreiber, B. Wielinga, and M. Worring. Classification of user image descriptions. *International Journal of Human-Computer Studies*, 61:601–626, Nov 2004.

[92] P. Jacquet, W. Szpankowski, and I. Apostol. A universal predictor based on pattern matching. *IEEE Transactions on Information Theory*, 48(6):1462 –1472, Jun 2002.

[93] A. Jaimes and S. Chang. A conceptual framework for indexing visual information at multiple levels. In *Proc. IS&T/SPIE Internet Imaging*, volume 3964, pages 2–15. SPIE, 2000.

[94] B. Jansen. Searching for digital images on the web. *Journal of Documentation*, 64:81–101, 2008.

[95] H. Jeung, Q. Liu, H. Shen, and X. Zhou. A hybrid prediction model for moving objects. In *Proc. IEEE Int'l Conf. Data Engineering*, pages 70–79. IEEE, 2008.

[96] Y. Jia, J. Wang, G. Zeng, H. Zha, and X. Hua. Optimizing kd-trees for scalable visual descriptor indexing. In *Proc. IEEE Conf. Computer Vision and Pattern Recognition (CVPR 10)*, pages 3392 –3399, 2010.

[97] T. Joachims. Optimizing search engines using clickthrough data. In *Proc. ACM SIGKDD Int'l Conf. Knowledge Discovery and Data Mining (KDD 02)*, pages 133–142. ACM, 2002.

[98] T. Joachims, L. Granka, B. Pan, H. Hembrooke, and G. Gay. Accurately interpreting clickthrough data as implicit feedback. In *Proc. Int'l ACM SIGIR Conf. Research and Development in Information Retrieval (SIGIR 05)*, pages 154–161. ACM, 2005.

[99] T. Joachims, L. Granka, B. Pan, H. Hembrooke, F. Radlinski, and G. Gay. Evaluating the accuracy of implicit feedback from clicks and query reformulations in web search. *ACM Transactions on Information Systems*, 25:7, 2007.

[100] R. Jones, R. Kumar, B. Pang, and A. Tomkins. "I know what you did last sum-

mer": query logs and user privacy. In *Proc. ACM Conf. Information and Knowledge Management*, CIKM '07, pages 909–914, 2007.

[101] C. Jorgensen. *Image Retrieval: Theory and Research*. Scarecrow Press, 2003.

[102] A. Karbassi and M. Barth. Vehicle route prediction and time of arrival estimation techniques for improved transportation system management. In *Proc. IEEE Intelligent Vehicles Symp.*, pages 511–516. IEEE, 2003.

[103] D. Katsaros and Y. Manolopoulos. Prediction in wireless networks by markov chains. *IEEE Wireless Communications*, 16(2):56–64, Apr 2009.

[104] R. Kenett and S. Salini. Relative linkage disequilibrium: A new measure for association rules. In P. Perner, editor, *Advances in Data Mining. Medical Applications, E-Commerce, Marketing, and Theoretical Aspects*, volume 5077 of *Lecture Notes in Computer Science*, pages 189–199. Springer, 2008.

[105] M. Khamsi and W. Kirk. *An introduction to metric spaces and fixed point theory*. Wiley-Interscience, 2001.

[106] J. Kim. Estimating classification error rate: Repeated cross-validation, repeated hold-out and bootstrap. *Computational Statistics and Data Analysis*, 53(11):3735 – 3745, 2009.

[107] R. Kohavi. A study of cross-validation and bootstrap for accuracy estimation and model selection. In *Proc. Int'l joint Conf. Artificial intelligence*, volume 2, pages 1137–1143. Morgan Kaufmann, 1995.

[108] T. Kollar. Fast nearest neighbors. Technical, Massachusetts Institute of Technology, 2006.

[109] J. Krumm. Ubiquitous advertising: The killer application for the 21st century. *IEEE Pervasive Computing*, 10(1):66 –73, Jan–Mar 2011.

[110] J. Kubica. *Efficient Discovery of Spatial Associations and Structure*. PhD thesis, Carnegie Mellon University, Pittsburgh, Pennsylvania, December 2005.

[111] J. Kubica, A. Moore, A. Connolly, and R. Jedicke. Spatial data structures for efficient trajectory-based queries. Technical report, Carnegie Mellon University, 2004.

[112] R. Kumar, J. Novak, B. Pang, and A. Tomkins. On anonymizing query logs via

token-based hashing. In *Proc. Int'l Conf. World Wide Web (WWW 07)*, pages 629–638. ACM, 2007.

[113] S. Lallich, O. Teytaud, and E. Prudhomme. Association rule interestingness: Measure and statistical validation. In F. Guillet and H. Hamilton, editors, *Quality Measures in Data Mining*, volume 43 of *Studies in Computational Intelligence*, pages 251–275. Springer, 2007.

[114] P. Lamb and S. Thiébaux. Avoiding explicit map-matching in vehicle location. In *Proc. World Conf. Intelligent Transportation Systems (ITS-99)*, 1999.

[115] D. Lee and C. Wong. Worst-case analysis for region and partial region searches in multidimensional binary search trees and balanced quad trees. *Acta Informatica*, 9:23–29, 1977.

[116] J. Letchner, J. Krumm, and E. Horvitz. Trip router with individualized preferences (trip): Incorporating personalization into route planning. In *Proc. National Conf. Artifical Intelligence*, volume 21, page 1795. AAAI, 2006.

[117] E. Levina and P. Bickel. The earth mover's distance is the mallows distance: some insights from statistics. In *Proc. Int'l Conf. Computer Vision (ICCV 01)*, volume 2, pages 251–256. IEEE, 7–14 July 2001 2001.

[118] L. Liao. *Location-based Activity Recognition*. PhD thesis, University of Washington, 2006.

[119] L. Liao, D. Fox, and H. Kautz. Extracting places and activities from gps traces using hierarchical conditional random fields. *The International Journal of Robotics Research*, 26(1):119, 2007.

[120] L. Liao, D. Fox, and H. Kautz. Hierarchical conditional random fields for gps-based activity recognition. In S. Thrun, R. Brooks, and H. Durrant-Whyte, editors, *Robotics Research*, volume 28 of *Springer Tracts in Advanced Robotics*, pages 487–506. Springer Berlin, 2007.

[121] L. Liao, D. Patterson, D. Fox, and H. Kautz. Building personal maps from GPS data. *Annals of the New York Academy of Sciences*, 1093:249–265, December 2006.

[122] L. Liao, D. Patterson, D. Fox, and H. Kautz. Learning and inferring transportation routines. *Artificial Intelligence*, 171(5-6):311–331, Apr 2007.

[123] H. Lieberman. Letizia: An agent that assists web browsing. In *Proc. Int'l Joint*

Conf. Articial Intelligence (IJCAI 95), volume 14, pages 924–929, Montreal, Quebec, Canada, 1995.

[124] J. Liu, X. Fan, and Z. Qu. A new interestingness measure of association rules. In C. Ryan and M. Keijzer, editors, *Proc. Int'l Workshop Genetic and Evolutionary Computing*, pages 393 –397. IEEE, 2008.

[125] R. Lomax. *An introduction to statistical concepts*. Lawrence Erlbaum, 2007.

[126] W. Lou and H. Lu. Efficient prediction of web accesses on a proxy server. In *Proc. Int'l Conf. Information and Knowledge Management (CIKM 02)*, pages 169–176. ACM, 2002.

[127] S. Lu and V. Bharghavan. Adaptive resource management algorithms for indoor mobile computing environments. *Computer Communication Review*, 26:231–242, 1996.

[128] R. Madhavan and C. Schlenoff. Moving object prediction for off-road autonomous navigation. In *Proc. SPIE Aerosense Conference*, volume 5083, page 134. SPIE, 2003.

[129] C. Mallows. A note on asymptotic joint normality. *The Annals of Mathematical Statistics*, 43(2):508–515, 1972.

[130] Y. Manolopoulos, A. Nanopoulos, and Y. Theodoridis. *R-trees: Theory and Applications*. Springer Verlag, 2006.

[131] M. Markatou, H. Tian, S. Biswas, and G. Hripcsak. Analysis of variance of cross-validation estimators of the generalization error. *Journal of Machine Learning Research*, 6:1127–1168, Dec 2005.

[132] N. Marmasse and C. Schmandt. Location-aware information delivery with commotion. In P. Thomas and H. Gellersen, editors, *Handheld and Ubiquitous Computing*, volume 1927 of *Lecture Notes in Computer Science*, pages 361–370. Springer Berlin, 2000.

[133] R. Meinhold and N. Singpurwalla. Understanding the kalman filter. *The American Statistician*, 37(2):123–127, 1983.

[134] N. Merhav and M. Feder. Universal prediction. *IEEE Transactions on Information Theory*, 44(6):2124 –2147, Oct 1998.

[135] A. Miyazaki. Online privacy and the disclosure of cookie use: Effects on consumer trust and anticipated patronage. *Journal of Public Policy & Marketing*, 27(1):19–33, 2008.

[136] A. Molinaro, R. Simon, and R. Pfeiffer. Prediction error estimation: a comparison of resampling methods. *Bioinformatics*, 21(15):3301–3307, 2005.

[137] A. Monreale, F. Pinelli, R. Trasarti, and F. Giannotti. WhereNext: a location predictor on trajectory pattern mining. In J. Elder IV, F. Fogelman-Soulié, P. Flach, and M. Zaki, editors, *Proc. ACM SIGKDD Int'l Conf. Knowledge Discovery and Data Mining (KDD 09)*, pages 637–646. ACM, 2009.

[138] A. Moore. *Efficient Memory-based Learning for Robot Control*. PhD thesis, Computer Laboratory, University of Cambridge, Cambridge, UK, October 1990.

[139] M. Morzy. Mining frequent trajectories of moving objects for location prediction. In P. Perner, editor, *Machine Learning and Data Mining in Pattern Recognition*, volume 4571 of *Lecture Notes in Computer Science*, pages 667–680. Springer, 2007.

[140] MuToss coding team, G. Blanchard, T. Dickhaus, N. Hack, F. Konietschke, K. Rohmeyer, J. Rosenblatt, M. Scheer, and W. Werft. *mutoss: Unified multiple testing procedures*, 2010. R package version 0.1-4.

[141] C. Nadeau and Y. Bengio. Inference for the generalization error. *Machine Learning*, 52:239–281, 2003.

[142] A. Naftel and S. Khalid. Classifying spatiotemporal object trajectories using unsupervised learning in the coefficient feature space. *Multimedia Systems*, 12:227–238, 2006.

[143] G. Navarro-Arribas, V. Torra, A. Erola, and J. Castellá-Roca. User k-anonymity for privacy preserving data mining of query logs. *Information Processing & Management*, In Press, 2011.

[144] R. Neapolitan. *Learning bayesian networks*. Pearson Prentice Hall Upper Saddle River, NJ, 2004.

[145] S. Nutanong, E. Jacox, and H. Samet. An incremental hausdorff distance calculation algorithm. In *Proc. Int'l Conf. Very Large Data Bases (VLDB 11)*, volume 4, pages 506–517. VLDB Endowment, 2011.

[146] C. Oliveira, J. Kim, and T. Suda. An adaptive bandwidth reservation scheme

for high-speed multimedia wireless networks. *Selected Areas in Communications*, 16(6):858–874, 1998.

[147] D. Papadogkonas, G. Roussos, and M. Levene. Analysis, ranking and prediction in pervasive computing trails. In *Proc. Int'l Conf. Intelligent Environments (IET 08)*, pages 1 –8, 2008.

[148] P. Pathirana, A. Savkin, and S. Jha. Mobility modelling and trajectory prediction for cellular networks with mobile base stations. In *Proc. ACM Int'l Symp. on Mobile ad hoc Networking & Computing (MobiHoc 03)*, pages 213–221. ACM, 2003.

[149] D. Patterson, L. Liao, D. Fox, and H. Kautz. Inferring high-level behavior from low-level sensors. In A. Dey, A. Schmidt, and J. McCarthy, editors, *UbiComp 2003: Ubiquitous Computing*, volume 2864 of *Lecture Notes in Computer Science*, pages 73–89. Springer Berlin, 2003.

[150] D. Patterson, L. Liao, K. Gajos, M. Collier, N. Livic, K. Olson, S. Wang, D. Fox, and H. Kautz. Opportunity knocks: A system to provide cognitive assistance with transportation services. In N. Davies, E. Mynatt, and I. Siio, editors, *UbiComp 2004: Ubiquitous Computing*, pages 433–450. Springer Berlin, 2004.

[151] T. Pedersen. A simple approach to building ensembles of naive bayesian classifiers for word sense disambiguation. In *Proc. Conf. North American chapter of the Association for Computational Linguistics*, pages 63–69. Morgan Kaufmann, 2000.

[152] J. Pei, J. Han, B. Mortazavi-Asl, H. Pinto, Q. Chen, U. Dayal, and M. Hsu. Prefixspan: mining sequential patterns efficiently by prefix-projected pattern growth. In *Int'l Conf. Data Engineering (ICDE 01)*, pages 215–224. IEEE, 2–6 April 2001 2001.

[153] O. Pele and M. Werman. A linear time histogram metric for improved sift matching. In D. Forsyth, P. Torr, and A. Zisserman, editors, *Computer Vision ECCV 2008*, volume 5304 of *Lecture Notes in Computer Science*, pages 495–508. Springer, 2008.

[154] O. Pele and M. Werman. Fast and robust earth mover's distances. In *Proc. Int'l Conf. Computer Vision (ICCV 09)*, pages 460 –467. IEEE, 2009.

[155] B. Poblete, M. Spiliopoulou, and R. Baeza-Yates. Privacy-preserving query log

mining for business confidentiality protection. *ACM Transactions on the Web*, 4:10:1–10:26, Jul 2010.

[156] G. Poitras. More on the correct use of omnibus tests for normality. *Economics Letters*, 90(3):304 – 309, 2006.

[157] T. Qin, T. Liu, X. Zhang, D. Wang, and H. Li. Global Ranking Using Continuous Conditional Random Fields. In D. Koller, D. Schuurmans, Y. Bengio, L. Bottou, D. Koller, D. Schuurmans, Y. Bengio, and L. Bottou, editors, *Proc. Advances in Neural Information Processing Systems (NIPS 08)*, pages 1281–1288. MIT Press, 2008.

[158] R Development Core Team. *R: A Language and Environment for Statistical Computing*. R Foundation for Statistical Computing, Vienna, Austria, 2010.

[159] V. Radosavljevic, S. Vucetic, and Z. Obradovic. Continuous Conditional Random Fields for Regression in Remote Sensing. In *Proc. European Conf. Artificial Intelligence (ECIA 10)*, pages 809–814. IOS Press, 2010.

[160] N. Ratliff, D. Silver, and J. Bagnell. Learning to search: Functional gradient techniques for imitation learning. *Autonomous Robots*, 27(1):25–53, July 2009.

[161] Y. Reich and S. Barai. Evaluating machine learning models for engineering problems. *Artificial Intelligence in Engineering*, 13(3):257 – 272, 1999.

[162] I. Rhee, M. Shin, S. Hong, K. Lee, S. Kim, and S. Chong. CRAWDAD data set ncsu/mobilitymodels (v. 2009-07-23). Downloaded from http://crawdad.cs.dartmouth.edu/ncsu/mobilitymodels, July 2009.

[163] C. Rijsbergen. *Information Retrieval*. Butterworths, 1979.

[164] J. Rissanen. A universal data compression system. *IEEE Transactions on Information Theory*, 29(5):656–664, Sep 1983.

[165] D. Ron, Y. Singer, and N. Tishby. The power of amnesia: Learning probabilistic automata with variable memory length. *Machine Learning*, 25:117–149, 1996.

[166] Y. Rubner, C. Tomasi, and L. Guibas. A metric for distributions with applications to image databases. In *Proc. Int'l Conf. Computer Vision (ICCV 98)*, pages 59–66, Los Alamitos, CA, 1998. IEEE.

[167] H. Samet. *Foundations of multidimensional and metric data structures*. Morgan Kaufmann, 2006.

[168] A. Sandham, T. Ormerod, C. Dando, R. Bull, M. Jackson, and J. Goulding. Scent trails: Countering terrorism through informed surveillance. In D. Harris, editor, *Engineering Psychology and Cognitive Ergonomics*, volume 6781 of *Lecture Notes in Computer Science*, pages 452–460. Springer Berlin, 2011.

[169] D. Sarkar. *Lattice: Multivariate Data Visualization with R*. Springer, New York, 2008.

[170] C. Schlenoff, R. Madhavan, and T. Barbera. A hierarchical, multi-resolutional moving object prediction approach for autonomous on-road driving. In *Proc. IEEE Int'l Conf. Robotics and Automation (ICRA 04)*, volume 2, pages 1956 – 1961, 2004.

[171] F. Scholer, M. Shokouhi, B. Billerbeck, and A. Turpin. Using clicks as implicit judgments: Expectations versus observations. In C. Macdonald, I. Ounis, V. Plachouras, I. Ruthven, and R. White, editors, *Advances in Information Retrieval*, volume 4956 of *Lecture Notes in Computer Science*, pages 28–39. Springer Berlin, 2008.

[172] T. Sellis, N. Roussopoulos, and C. Faloutsos. The R^+-tree: A dynamic index for multi-dimensional objects. In *Proc. Int'l Conf. Very Large Data Bases VLDB 87*, pages 507–518. Morgan Kaufmann, 1987.

[173] A. Sharma, G. Hua, Z. Liu, and Z. Zhang. Meta-tag propagation by co-training an ensemble classifier for improving image search relevance. In *Proc. IEEE Conf. Computer Vision and Pattern Recognition Workshops (CVPRW 08)*, pages 1–6. IEEE, 2008.

[174] S. Shatford. Analyzing the subject of a picture: A theoretical approach. *Cataloging and Classification Quarterly*, 6(3):39–61, 1986.

[175] D. Shen, M. Qin, W. Chen, Q. Yang, and Z. Chen. Mining web query hierarchies from clickthrough data. In *Proc. National Conf. Artificial Intelligence*, volume 22, page 341. AAAI, 2007.

[176] M. Shokouhi, F. Scholer, and A. Turpin. Investigating the effectiveness of clickthrough data for document reordering. In C. Macdonald, I. Ounis, V. Plachouras,

I. Ruthven, and R. White, editors, *Advances in Information Retrieval*, volume 4956 of *Lecture Notes in Computer Science*, pages 591–595. Springer Berlin, 2008.

[177] S. Sigg. *Development of a novel context prediction algorithm and analysis of context prediction schemes*. PhD thesis, University of Kassel, Germany, 2008.

[178] S. Sigg, S. Haseloff, and K. David. An alignment approach for context prediction tasks in ubicomp environments. *IEEE Pervasive Computing*, 9:90–97, 2010.

[179] C. Silpa-Anan and R. Hartley. Optimised kd-trees for fast image descriptor matching. In *Proc. IEEE Conf. Computer Vision and Pattern Recognition (CVPR 08)*, pages 1 –8, 2008.

[180] R. Simmons, B. Browning, Y. Zhang, and V. Sadekar. Learning to predict driver route and destination intent. In *Proc. Intelligent Transportation Systems Conf. (ITSC 06)*, pages 127–132. IEEE, 2006.

[181] G. Smith and H. Ashman. Evaluating implicit judgements from image search interactions. In *Proc. Web Science Conf.: Society On-Line (WebSci 09)*, 2009.

[182] L. Song, D. Kotz, R. Jain, and X. He. Evaluating next-cell predictors with extensive wi-fi mobility data. *IEEE Transactions on Mobile Computing*, 5:1633–1649, 2006.

[183] R. Sriraghavendra, K. Karthik, and C. Bhattacharyya. Fréchet distance based approach for searching online handwritten documents. In *Int'l Conf. Document Analysis and Recognition (ICDAR 07)*, volume 1, pages 461 –465, 2007.

[184] P. Tan, V. Kumar, and J. Srivastava. Selecting the right objective measure for association analysis. *Information Systems*, 29:293–313, 2004.

[185] K. Torkkola, K. Zhang, H. Li, H. Zhang, C. Schreiner, and M. Gardner. Traffic Advisories Based on Route Prediction. In *Proc. Workshop Mobile Interaction with the Real World*, page 33. SAE, 2007.

[186] M. Truran, J. Goulding, and H. Ashman. Co-active intelligence for image retrieval. In *Proc. ACM Int'l Conf. Multimedia (MM 05)*, pages 547–550. ACM, 2005.

[187] J. Tsai, P. Kelley, L. Cranor, and N. Sadeh. Location-sharing technologies: Privacy risks and controls. *A Journal of Law & Policy for the Information Society*, 6:119–317, 2010.

[188] J. Uhlmann. Satisfying general proximity / similarity queries with metric trees. *Information Processing Letters*, 40(4):175 – 179, 1991.

[189] C. Vaitl, K. Kunze, and P. Lukowicz. Does on-body location of a GPS receiver matter? In *Proc. Int'l Conf. Body Sensor Networks (BSN 10)*, pages 219 –221, 2010.

[190] M. Vassilakopoulos, A. Corral, and N. Karanikolas. Join-queries between two spatial datasets indexed by a single R*-tree. In I. Cerná, T. Gyimóthy, J. Hromkovic, K. Jefferey, R. Královic, M. Vukolic, and S. Wolf, editors, *SOFSEM 2011: Theory and Practice of Computer Science*, volume 6543 of *Lecture Notes in Computer Science*, pages 533–544. Springer Berlin, 2011.

[191] P. Volf. *Weighting Techniques in Data Compression: Theory and Algorithms*. PhD thesis, Technical University of Eindhoven, 2002.

[192] I. Wald and V. Havran. On building fast kd-trees for ray tracing, and on doing that in o(n log n). In *Proc. IEEE Symp. Interactive Ray Tracing*, pages 61 –69. IEEE, 2006.

[193] G. Webb. OPUS: an efficient admissible algorithm for unordered search. *Journal of Artificial Intelligence Research*, 3:431–465, 1995.

[194] J. Wen, J. Nie, and H. Zhang. Clustering user queries of a search engine. In *Proc. Int'l Conf. World Wide Web (WWW 01)*, pages 162–168. ACM, 2001.

[195] C. Wenk, R. Salas, and D. Pfoser. Addressing the need for map-matching speed: Localizing global curve-matching algorithms. In *Int'l Conf. Scientific and Statistical Database Management*, volume 0, pages 379–388. IEEE, 2006.

[196] F. Willems, Y. Shtarkov, and T. Tjalkens. The context-tree weighting method: basic properties. *IEEE Transactions on Information Theory*, 41(3):653 –664, May 1995.

[197] C. Willmott and K. Matsuura. Advantages of the mean absolute error (MAE) over the root mean square error (RMSE) in assessing average model performance. *Climate Research*, 30(1):79, 2005.

[198] I. Witten and T. Bell. The zero-frequency problem: estimating the probabilities of novel events in adaptive text compression. *IEEE Transactions on Information Theory*, 37(4):1085 –1094, July 1991.

[199] D. Wuertz and Rmetrics core team members. *fBasics: Rmetrics - Markets and Basic Statistics*, 2010. R package version 2110.79.

[200] G. Xue, Z. Li, H. Zhu, and Y. Liu. Traffic-Known urban vehicular route prediction based on partial mobility patterns. In *Proc. Int'l Conf. Parallel and Distributed Systems*, pages 369–375. IEEE, 2009.

[201] G. Xue, H. Zeng, Z. Chen, Y. Yu, W. Ma, W. Xi, and W. Fan. Optimizing web search using web click-through data. In *Proc. ACM Int'l Conf. Information and Knowledge Management (ICIKM 04)*, pages 118–126. ACM, 2004.

[202] Q. Ye, L. Chen, and G. Chen. Predict personal continuous route. In *Proc. Int'l Conf. Intelligent Transportation Systems*. IEEE, 2008.

[203] V. Zheng, Y. Zheng, X. Xie, and Q. Yang. Collaborative location and activity recommendations with GPS history data. In *Proc. Int'l Conf. World Wide Web (WWW 10)*, pages 1029–1038. ACM, 2010.

[204] F. Zhong, D. Wang, G. Wang, W. Chen, Y. Zhang, Z. Chen, and H. Wang. Incorporating post-click behaviors into a click model. In *Proc. ACM SIGIR Conf. Research and Development in Information Retrieval (SIGIR 10)*, pages 355–362, 2010.

[205] B. Ziebart, N. Ratliff, G. Gallagher, C. Mertz, K. Peterson, J. Bagnell, M. Hebert, A. Dey, and S. Srinivasa. Planning-based prediction for pedestrians. In *Proc. IEEE/RSJ Int'l Conf. Intelligent Robots and Systems (IROS 09)*, pages 3931–3936. IEEE, 2009.

[206] D. Zimmerman. A note on interpretation of the paired-samples t test. *Journal of Educational and Behavioral Statistics*, 22(3):349–360, 1997.

Appendix A

R code: the truncated average discrete Fréchet distance

Listing A.1: A function for the **R** statistical computing software [158] to calculate the Average Fréchet Distance. Based on the *pathFrechet* function released under the GPL licence ≥ 2 in the package *longitudinalData* [72]. The modified function is also released under the GPL licence ≥ 2.

```
avgFrechetDist <-function(P, Q, Fdist = dist,
                          penalizeOverpredict = FALSE, autoResample = TRUE )
{

  if( autoResample ){
    if( length(P) != 1 && length(Q) != 1 ){
      resampleRate = min( getSmallestSampleRate( P ),
                          getSmallestSampleRate( Q ) )
      if( resampleRate == 0 ) {
        stop( 'Error: Duplicate points in input.' )
      }
      P = resample(P, resampleRate, Fdist )
      Q = resample(Q, resampleRate, Fdist )
    } else if( length(P) == 1 ){
      resampleRate = getSmallestSampleRate( Q )
      if( resampleRate == 0 ) {
        stop( 'Error: Duplicate points in input.' )
      }
      Q = resample(Q, resampleRate, Fdist )
    } else if( length(Q) == 1 ){
      resampleRate = getSmallestSampleRate( P )
```

```
    if( resampleRate == 0 ) {
      stop( 'Error: Duplicate points in input.' )
    }
    P = resample(P, resampleRate, Fdist )
  }

}
method = "sum"
if (identical(Fdist, "2D")) {
  Fdist <- dist
}
way <- c("PQ", "P", "Q")
maxP <- length(P)
maxQ <- length(Q)
Mpath <- Mdist <- Mfret <- matrix(0, maxP, maxQ,
                  dimnames = c(list(paste("P", 1:maxP, sep = "")),
                  paste("Q", 1:maxQ, sep = ""))))
for (i in 1:maxP) {
  for (j in 1:maxQ) {
    Mdist[i, j] <- Fdist(rbind(P[[i]], Q[[j]]))
    if (i == 1 && j == 1) {
      Mfret[1, 1] = Mdist[1, 1]
      Mpath[1, 1] = "start"
    }
    if (i > 1 && j == 1) {
      Mfret[i, 1] = do.call(method, list(Mfret[i -
                    1, 1], Mdist[i, 1]))
      Mpath[i, 1] = "P"
    }
    if (i == 1 && j > 1) {
      Mfret[1, j] = do.call(method, list(Mfret[1, j -
                    1], Mdist[1, j]))
      Mpath[1, j] = "Q"
    }
    if (i > 1 && j > 1) {
      movePQ <- Mfret[i - 1, j - 1]
      moveP <- Mfret[i - 1, j]
      moveQ <- Mfret[i, j - 1]
      Mfret[i, j] = do.call(method, list(min(movePQ,
                    moveP, moveQ), Mdist[i, j]))
      Mpath[i, j] = way[which.min(c(movePQ, moveP,
```

```
                    moveQ))]
    }
  }
}
i <- maxP
j <- maxQ
bestPath <- c(maxP, maxQ)
while (i > 1 || j > 1) {
  if (Mpath[i, j] == "Q") {
    j <- j - 1
  }
  else {
    if (Mpath[i, j] == "P") {
      i <- i - 1
    }
    else {
      i <- i - 1
      j <- j - 1
    }
  }
  bestPath <- rbind(c(i, j), bestPath)
}
colnames(bestPath) <- c("P", "Q")
rownames(bestPath) <- NULL
distToRtn = Mfret[maxP, maxQ]
if( !penalizeOverpredict ){
  tmp = bestPath[,2]
  idxs = which( tmp == length(Q) )
  m = idxs[1]
  # If there are more that one segment matched
  # to the last in the ground truth
  if( length(idxs) > 1 ){
    removeIdxs = idxs[ 2:(length(idxs))]
    amountToRemove = 0
    for( i in removeIdxs ){
      idxToRm = bestPath[i,]
      amountToRemove = amountToRemove + as.numeric(
                  Fdist( rbind( P[[ as.numeric( idxToRm[ 1 ] ) ]],
                  Q[[ as.numeric( idxToRm[ 2 ] ) ]] )) )
    }
    distToRtn = distToRtn - amountToRemove
```

```
    }
  }
  if( penalizeOverpredict ){
    distToRtn <-distToRtn / dim(bestPath)[1]
  } else {
    distToRtn <-distToRtn / m
  }
  return(list(dist = distToRtn, bestPath = bestPath))
}
```

Listing A.2: Function to determine the smallest distance between two points in a sequence of sample points.

```
getSmallestSampleRate <- function( listIn ){
  first = 1:(length( listIn )-1)
  second = 2:length( listIn )
  minDistance = 999999999 #A large number
  for( i in first ){
    distance = dist( rbind( listIn[[ first[ i ] ]],
                     listIn[[ second[ i ] ]] ) )
    if( distance < minDistance ){
      minDistance = distance
    }
  }
  return( minDistance )
}
```

Listing A.3: Function to resample a sequence given a fixed sample rate.

```
resample <- function( listIn , sampleRate, distF ){
  toRtn = list()
  if( length( listIn ) == 0 ) {
    stop("Can not resample a path with no points!")
  }
  if( length( listIn ) == 1 ) return( listIn )
  segStart = listIn[[1]]
  toRtn = list( segStart )
  remainder = 0
  for( i in 2:length(listIn) ){
    segEnd = listIn[[ i ]] ;
    segDeltaX = segEnd[ 1 ] - segStart[ 1 ];
```

```
    segDeltaY = segEnd[ 2 ] - segStart[ 2 ]
    segLen = distF( rbind( segStart, segEnd ) )
    moveForwardBy = sampleRate - remainder
    remainder = segLen;
    if( is.infinite( segLen ) ||
        is.nan( segLen ) ||
        is.infinite( moveForwardBy ) ||
           is.nan( moveForwardBy ) ||
            moveForwardBy == 0 ){
      stop( "Error: Invalid values detected." )
    }
    while( segLen / moveForwardBy >= 1 ){
      percent = moveForwardBy / segLen;
      deltaX = segDeltaX * percent;
      deltaY = segDeltaY * percent;
      newX = segStart[ 1 ] + deltaX;
      newY = segStart[ 2 ] + deltaY;
      toRtn[[ (length( toRtn ) + 1 ) ]] <- c( newX, newY )
      remainder = segLen - moveForwardBy
      moveForwardBy = moveForwardBy + sampleRate
    }
    segStart = segEnd
  }
  if( length(toRtn) == 0 ) {
    stop( "Resampling failed - no points returned." );
  }
  return( toRtn )
}
```