Parameter Optimisation for Location Extraction and Prediction Applications

Alasdair Thomason, Nathan Griffiths, Victor Sanchez
Department of Computer Science
University of Warwick, Coventry
United Kingdom, CV4 7AL
{ali, nathan, vsanchez}@dcs.warwick.ac.uk

***Abstract*—The pervasive nature of location-aware hardware has provided an unprecedented foundation for understanding human behaviour. With a record of historic movement, in the form of geospatial trajectories, extracting locations meaningful to users is commonly performed as a basis for modelling a users' interactions with their environment. Existing literature, however, has scarcely considered the applicability of extracted locations, typically focusing solely on the consequent knowledge acquisition process employed, due to the difficulty of evaluating the output of such unsupervised learning techniques. Towards the goal of ensuring the representativeness of extracted locations, and using location prediction as an example knowledge acquisition process, this work provides a method of automated parameter selection for both location extraction and prediction that ensures both the applicability of the locations extracted and the utility of the predictions performed. Specifically, we: (i) provide a metric for the evaluation of both extracted locations and predictions that characterises the goal of each of these tasks, (ii) frame the process of parameter selection as that of mathematical optimisation through the presented metric, and (iii) discuss characteristics of the metric while demonstrating its applicability over real-world data, location extraction algorithms and prediction techniques.**

I. Introduction

With vast amounts of geospatial data readily available, services are increasingly being built to leverage the latent knowledge inherent in human mobility patterns. Understanding human behaviour through data has great potential in improving the quality and personalisation of services offered, making for tailored solutions capable of understanding not only the current needs and preferences of its users, but also their likely future requirements. Many systems that focus on understanding people from geospatial data make use of location extraction techniques to identify locations that have meaning and importance to the users, gaining insight not only into the types of places that users spend their time, but also aiding in the identification of the activities they perform and the people they associate with. These extracted locations are often used as the basis for the prediction of future movements of individuals, the recommendation of places to visit, and the identification of activities conducted.

While the extraction of locations meaningful to users provides a vital foundation for further analysis, existing literature has scarcely considered the applicability of the locations extracted to the task at hand, instead focusing solely on the consequent knowledge acquisition process employed. Although evaluating locations extracted through unsupervised learning techniques is challenging due to the lack of available ground truth, the properties inherent in the locations have significant impact on the results of further analysis. Selecting appropriate parameters for location extraction is therefore of paramount importance. However, the impact of altering parameters is typically unknown, as methods of evaluating the applicability of a set of locations to a particular task are lacking.

The work presented here aims to overcome this problem by providing a method of parameter optimisation that selects the most appropriate parameters for both location extraction and the subsequent learning algorithm employed (e.g. location prediction). The need for manual selection of parameters and review of location properties is therefore removed, and the robustness of any predications based on the locations is ensured, thereby increasing the utility afforded by such systems. Specifically, we focus on the problem of location prediction as a representative task and (i) provide a metric for the evaluation of both extracted locations and predictions that characterises the goal of each of these tasks, (ii) frame the process of parameter selection as that of mathematical optimisation through the presented metric, and (iii) discuss characteristics of the metric while demonstrating its applicability over real-world data, location extraction algorithms and prediction techniques.

The remainder of this paper is structured as follows: Section II discusses related work in the areas of location extraction, prediction and parameter optimisation. Section III introduces the proposed parameter optimisation procedure for the task of location extraction and prediction. Section IV presents the experimental methodology followed, and Section V presents and discusses the results observed when applying the methodology to real-world data. We discuss our conclusions in Section VI.

II. Related Work

This section discusses relevant existing literature for location extraction, prediction and parameter optimisation.

A. Location Extraction

Location extraction can be performed using a single clustering technique to extract areas of dense data points, however, it is typically split into two distinct steps [2, 4, 20, 30]. The first step, referred to as *visit extraction* or *stay point detection*, is concerned with detecting periods of low mobility in geospatial data, representing periods of time in which the individual remained stationary. This process is typically conducted in linear time. The second step, *visit clustering*, clusters these periods, henceforth referred to as *visits*, into locations.

In support of visit extraction, methods have been proposed to identify periods of missing data, where missing data is expected to correlate with a user being indoors [3, 4]. Recent work, however, has assumed a greater availability of data and consequent removal of the requirement for visits to only occur inside buildings. This work has focused on using time and distance thresholds [2, 18, 20, 31, 32], where the user must have remained for a specified period of time in an area no larger than a provided radius.

While these algorithms do not assume that missing data is synonymous with a visit, they all suffer from a lack of resilience to noise, as a single erroneous point that falls outside of the distance threshold causes the visit to be ended. Based on the fact that the majority of location estimation techniques are prone to noise, the STA extraction [7] and GVE algorithms [26] have been proposed. Both algorithms monitor the trend of motion of the user and employ techniques to ensure resilience to noise. GVE, the Gradient-based Visit Extractor, in contrast to STA extractor, performs visit extraction in such a way that it does not impose a minimum bound on visit duration and does not assume evenly spaced observations [26].

Once extracted, visits are grouped to form locations using clustering techniques, such as k-means [4, 19]. The drawback to this approach is the requirement for the number of clusters to be known a priori, and so the use of density-based approaches such as DBSCAN is more common [2, 6, 12, 20].

B. Location Prediction

Originally motivated by the problem of determining the future location of an individual in a cellular network, location prediction aims to identify which location a user will visit next given their current location and transition history. Many approaches have been applied to next location prediction, including neural networks [1, 9], support vector machines (SVMs) [28], and Markov models [4, 5, 14, 15]. These techniques have considered locations modelled as cellular towers [1, 9, 14], rooms in a smart office building [22, 27] and locations extracted through clustering approaches [4, 5, 10, 15]. The locations can be used as nodes in Markov models of transitions, and neural networks and SVMs can be used as classifiers for locations.

C. Parameter Optimisation

While much research has been conducted into location extraction and prediction, there has been little discussion of the selection of appropriate parameters for these techniques, with existing work using empirically determined parameters to demonstrate results. An exception to this is Ashbrook and Starner's presented method of parameter selection, which involves plotting graphs of the number of locations extracted for different parameters and selecting an appropriate value by manually observing a specific quality on the graph [3, 4].

Automated selection of parameters has, however, been considered for a few select domains, specialised for each task at hand. This includes optimising the parameters for support vector machines using online Gaussian process models [13], genetic algorithms [25] and heuristic-based approaches [23]. Also considered for parameter optimisation have been hidden Markov models, using length modelling [33], and neural networks, using genetic algorithms [11, 29].

Not focusing specifically on parameter selection, several mathematical optimisation and search algorithms have been presented that aim to locate an optimal or near-optimal solution from an n-dimensional search space. One such example is *hill climbing*, which begins at a random point in the search space and repeatedly moves to adjacent states until it reaches a maxima [24]. There is no guarantee, however, that such a maxima would be global as hill climbing is prone to detecting local maxima in non convex search spaces. Overcomming this issue, heuristic-based approaches have been presented including simulated annealing [8] and evolutionary approaches such as genetic algorithms [24], and memetic algorithms [21]. These algorithms do not guarantee to find the global maxima, but find an approximation of it given sufficient time to search. In this paper we describe how simulated annealing can be applied to the problem of parameter optimisation for location extraction and prediction.

III. Parameter Optimisation

Identifying an optimal, or near-optimal, set of parameters for both location extraction and prediction can be performed through mathematical optimisation techniques only once an evaluative procedure or metric has been defined. It is through such a metric that two sets of parameters can be compared to identify the more desirable. The task of parameter optimisation can therefore be split into two primary components; an evaluative metric that, when given two sets of parameters, is able to quantifiably select the more desirable, and a mathematical optimisation procedure that, when combined with the evaluative metric, locates a near-optimal set of parameters in finite time. The remainder of this section details the development of the evaluative metric and the selection of an appropriate mathematical optimisation procedure for this purpose.

A. Evaluation Metric

In order to evaluate the combined performance of location extraction and prediction, it becomes important to understand the true aims behind the process. For this work, we consider the aim of location prediction to be that of identifying the exact future location of an individual with as little uncertainty as possible. Uncertainty in this context can be considered to encompass both the size of locations and the accuracy of predictions. Considering the example where extracted locations cover vast regions, it is clear that predictive accuracy would be fairly high as the task only requires the predictor to identify which location, e.g. city, the user will be in at a given time. Despite this high accuracy, the utility of the predictions would be low in many systems due to the size of the locations used. Conversely, if the locations extracted were particularly small (e.g. room-size) then the complexity of the user's behaviours present would be higher and thus more difficult to accurately predict, leading to lower predictive accuracy, but in cases where the prediction were correct, the utility afforded would be far greater. It is even conceivable that, when aiming to identify the exact geographic region a user will visit, a prediction for a small, close-by, location that is incorrect may offer greater knowledge than a prediction for a vast location that encompasses the correct region, as the former case, although incorrect, identifies a position close to the correct one.

Evaluating both locations and predictions is difficult in standard approaches because of the dependency relationship between the two. The exact locations extracted will directly impact on the ability for prediction to occur, and thus methods of characterising the locations and predictions independently cannot cater for this relationship. Therefore, the separate stages of location extraction and prediction must be evaluated together to get an honest representation of applicability to the given task. Towards this end, and taking the goal of location prediction to be that previously defined, the idea of aiming to identify as close as possible the region on earth to be visited by the user, we define *error* as follows.

Definition 1: The *error* of location extraction and prediction is the distance between the centroid (i.e. the arithmetic mean of all of its points) of the predicted location and the centroid of the region actually visited by the user, represented by the next visit in the data.

Intuitively, this definition of error favours small locations with accurate predictions, as wherever the actual region visited falls within such a location (i.e. an accurate prediction), the distance between the region and the centre of the location will be small. With large locations, which are undesirable for location prediction, a correct prediction may still have a high error if the distance between the location centroid and actual visited region is vast. Similarly, for incorrect predictions (i.e. inaccurate ones), a small predicted location situated near to the correct region will give a low error as the distance between the predicted location's centroid and the actual visit made is small.

Under such a definition of error, while small locations are favoured, locations that are meaninglessly small (e.g. if every data point were to be classed as its own location) are prevented by the properties inherent in predictive systems. In order to predict future movements of individuals, past behaviour is analysed and patterns determined, but when considering such an extreme case, each location would have a single transition to another, unique, location and thus no repeating patterns can exist. This property ensures that accurate predictions from such training data are unachievable and therefore predictions will be little better than random, producing a high average error in the system. This ensures that the locations favoured by Definition 1 are small, while still being meaningful. If the error were defined to be the distance between the centroid of the predicted location and the extracted location within which the user's next visit falls, a correct prediction would be given an error of 0 regardless of the size of the location. By using the distance between the centroids of the predicted location and the actual visit, unless the location covers only this single visit, the error will be non-zero even for a correct prediction, with a magnitude dependent upon the location's size.

The error of a given set of locations and given predictor, E, can be calculated by using the *mean absolute error* metric, with distances calculated by the haversine formula in kilometers, formally:

$$E = \frac{1}{|P|} \sum_{l,v \in P} |dist(centroid(l), centroid(v))| \qquad (1)$$

where P is the set of all predictions, each prediction having two parts: $\{l, v\} \in P$, l is the expected next location of the user, and v is the actual visit that the user makes next. Mean absolute error (MAE) is selected for its linear weighting of errors. While mean squared error is also a common metric, any large error would be weighted extremely highly and overshadow the remaining data (e.g. if the user visited a different city to the one predicted just once, even if all other predictions were correct), which is undesirable in this case. With MAE, a small number of large errors has significantly less impact and thus individual mistakes are still penalised, but not as highly. With MAE selected as the metric, and a definition of error (Definition 1) consistent with expectations of the problem, comparison of sets of extracted locations and predictions can occur. Given two sets of locations and an associated prediction model, the more desirable set is the one which has the lowest associated cost as calculated by MAE.

B. Optimisation Algorithm

With an evaluative metric in place, the selection of optimal parameters for a given set of data would ideally be performed by evaluating all possible combinations of parameters and selecting those which produce the minimum mean absolute error. In reality, however, performing location extraction and prediction is computationally expensive, and so it is infeasible to perform a complete search. Instead, a near optimal solution can be found using techniques including heuristic-based approaches (e.g. simulated annealing) or evolutionary approaches (e.g. genetic algorithms). Specifically, we opt to use simulated annealing as it can overcome the problem of local maxima while maintaining a single state space. While other algorithms, such as evolutionary approaches, are also applicable, they typically assume that taking two states that individually produce good results and merging them will produce results at least as good. In this work, the interplay between parameters is extremely important and thus it is not immediately clear that this property will hold, as such, an exploration of the applicability of different optimisation techniques is left as future work.

Simulated annealing starts at a random point in the search space (the parameter space in our case), evaluates the current position and selects a single-step neighbour (i.e. a parameter set that can be reached by modifying a single parameter by a single increment). If evaluating this neighbour yields a more favourable position than the current location, the move is taken. Otherwise, the move is taken with some probability according to the *temperature function* and *probability function*, together regulating the moves made. This selection procedure is repeated a maximum of k_{max} times, where k_{max} is a user-specified parameter. If there is plenty of time remaining, i.e. the iteration count, k, is low with respect to k_{max}, a bad move is more likely to be followed as there is room to recover later. If time is running out (k is approaching k_{max}), then only positive moves are likely to be made. Once a run has been completed, the parameters selected are those that produced the lowest score encountered throughout the optimisation procedure, typically the last move.

IV. Experimental Methodology

In order to explore the applicability of the proposed metric and algorithm to the problem of parameter selection for location extraction and prediction, several experiments must be conducted using different extraction and prediction approaches.

A. Location Extraction

As discussed in Section II-A, location extraction is typically performed as a two-step process where first periods of low mobility, called *visits*, are detected in linear time from the dataset. These visits are then clustered together into *locations*. For this work, we adopt two approaches to visit extraction, namely the *stay-point* detection method proposed in [18] that uses maximum size and minimum duration thresholds, and the gradient-based visit extractor algorithm proposed in [26][1], referred to as *thresholding* and *GVE* respectively. Thresholding takes two parameters, $radius$ and $time$, controlling the maximum size and minimum duration of a visit. GVE takes four parameters, α, β, n_{points} and t_{max}, with α and β controlling the threshold function that dictates when a visit is marked as having ended. Additionally, n_{points} sets the maximum size of buffer over which to consider trend of motion, and t_{max} specifies the maximum period of time between consecutive points for the points to belong to the same visit. This approach protects against periods of missing data from erroneously being grouped into the same visit where the user could have left and returned to a nearby location. While the thresholding approach is perhaps the most widely used in literature, the GVE algorithm is designed to overcome many of its drawbacks, and hence both are used to explore the generality of our approach. Once extracted, the resultant visits are then clustered using DBSCAN. DBSCAN has been demonstrated to be applicable to the problem and does not require the number of locations to be known a priori, instead determining the clusters based on density of visits, determined by the parameters $minpts$ and ϵ, which specify the minimum number of points required within distance ϵ to consider a cluster. The parameters for these algorithms are shown in Table I.

B. Location Prediction

While many methods of location prediction have been considered (Section II-B), for this work we focus specifically on the task of next location prediction, i.e. aiming to identify the location out of the extracted set of locations that the user is most likely to visit upon leaving their current location. For this purpose, we employ hidden Markov models and multilayer perceptions trained through backpropogation with a single hidden layer. Both of these approaches have been widely used for a range of tasks, including location prediction, and have been shown to be effective [1, 9, 14, 15]. Although alternative methods exist, such as SVMs, this work focuses on the proposed evaluative metric and a comparison of alternative predictors is left as future work.

Once a set of locations has been extracted, the history of transitions between them is split into two with half used to train the predictive models, and the remaining half used as part of the evaluation procedure. To predict future locations from historic transitions, the test data must be temporally after the training data, meaning that validation methods such as cross-validation are not appropriate. An even test:train split was selected to ensure both sufficient training data for the predictive models and to evaluate performance. Both of these approaches take parameters that can be optimised to improve predictive accuracy, shown later in Table I. For the hidden Markov model, this is the number of hidden states, *numStates*, and for the neural network it is the number of nodes in the hidden layer, *numNodes*. In addition, we introduce another parameter for the hidden Markov model, namely the sequence length, *seqLength*, to provide to the model when requesting a prediction. A long sequence length captures the recent movements of the user better, but affords fewer opportunities for prediction as at least *seqLength* transitions are required for prediction to occur. The selection of an appropriate value is therefore left open for optimisation as part of this process.

C. Parameter Optimisation

As described in Section III, the task of parameter optimisation is being performed using the Simulated Annealing algorithm in conjunction with a novel evaluative metric (Section III-A). The maximum number of moves that can be made by the Simulated Annealing algorithm is set by the parameter k_{max}, and thus bounds the time complexity. Complexity is, however, not linear in our case as the evaluative procedure requires both locations to be extracted and a predictive model to be trained, and thus complexity is dependant upon the specific algorithms used for these purposes. However, the increased complexity of such algorithms is somewhat mitigated as location extraction and prediction is only performed over a subset of the data.

Neighbour Function

In order to utilise simulated annealing to optimise the parameters for location extraction and prediction, a method of selecting a *neighbour* of a current state must be specified. As many of the parameters are continuous, we first discretise them by specifying a minimum increment value for each parameter as shown in Table I. The starting position of the algorithm is created by selecting a value for each parameter that falls within the start range, as shown in the table. We can now define the neighbour of a given state as being any set of parameters that has a single parameter altered by one increment such that the parameter combination is legal (i.e. all constraints imposed on the parameters by algorithms hold, for example, most parameters must be positive). The neighbour function of the simulated annealing algorithm selects one neighbour of the current state at random by adhering to these rules.

Probability Function

Once a neighbour has been selected, the probability function determines whether it will replace the current state, or if the state remains unchanged. Mean absolute error is used to provide a cost for both of the states. If the cost for the new state is lower, then the move is made. Additionally, if a cost cannot be calculated for the new state (e.g. it extracts no locations) then the neighbour is discarded and another selected. Finally,

[1] This algorithm has been extended with the addition of the parameter t_{max}, where a visit under consideration is ended if the time between the last point in the buffer and the new point is greater than t_{max} seconds, employed to prevent periods of missing data from being considered as part of a visit.

TABLE I: Parameter increments

Method	Algorithm	Parameter	Inc.	Start Range
Visit Extraction	GVE	α	0.1	$0 \leq \alpha \leq 2.5$
		β	3	$1 \leq \beta \leq 50$
		n_{points}	3	$1 \leq n \leq 50$
		t_{max} (s)	1200	$600 \leq t \leq 86400$
	Thresholding	*radius* (m)	10	$0 \leq r \leq 250$
		time (s)	120	$0 \leq t \leq 3600$
Visit Clustering	DBSCAN	*minpts*	1	$0 \leq m \leq 10$
		ϵ (m)	5	$1 \leq \epsilon \leq 100$
Prediction	HMM	*numStates*	1	$5 \leq n \leq 50$
		seqLen	1	$1 \leq s \leq 5$
	Neural Network	*numNodes*	1	$5 \leq n \leq 50$

a move to a worse position (i.e. higher cost) can be made with some probability dependant upon the *temperature*, which reduces with time. Initially, a worse move is more likely to be made to avoid being stuck in a local maxima while there is time to recover later. As time progresses, however, worse moves have a lower chance of being made.

To adhere to these principles, a probability function has been selected:

$$p(c,n,t) = \begin{cases} 1 & \text{if } n > c \\ 0 & \text{if } n = null \\ e^{-\frac{c-n}{t}} & \text{otherwise} \end{cases} \tag{2}$$

where c is the score of the current position, n is the score of the selected neighbour (which is set to *null* if no score can be calculated) and t is the current temperature. The probability of selecting a worse move depends upon the magnitude of the cost difference and the temperature, defined as:

$$t(r) = 0.985^{500r} \tag{3}$$

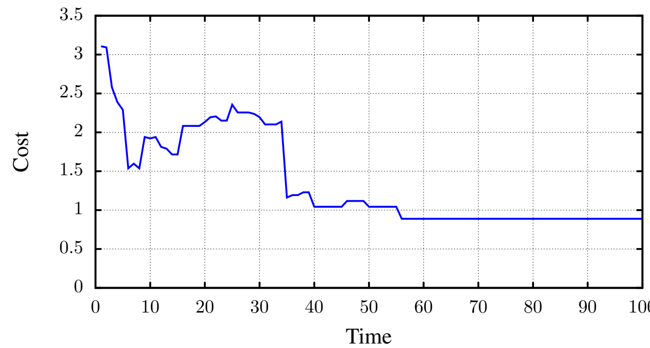
where r is the proportion of time expanded. The function ensures that temperature reduces rapidly towards 0 so that poor moves are only likely towards the beginning of the process. The exact values have been empirically determined to provide an appropriate function that responds to the problem at hand.

D. Data

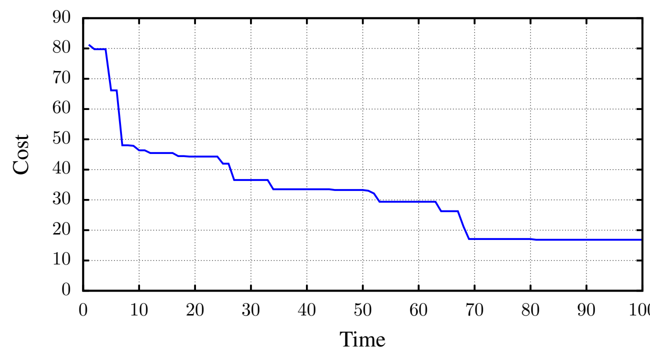
Our primary evaluation data comes from 5 users in the Nokia Mobile Data Challenge (MDC) Dataset [16, 17], which contains high-accuracy location and usage data collected from smartphones carried by 191 users over a period of 2 years.

E. Experiments

In order to understand the applicability of the metric proposed in Section III-A to the task at hand, multiple runs of the parameter optimisation approach must be performed with different combinations of location extraction and prediction algorithms and data from different users. Furthermore, to better understand the metric, it is important to investigate the impact of using different segments of data from the same users. To this end, experiments use both different amounts of data (continuous subsets of between 10% and 25% of the available data per user) with different starting positions (selected at points 0% and 50% through the data when ordered temporally). With data selected and a methodology formalised, experiments can be run with different values of k_{max}, the parameter of simulated annealing that specifies the maximum number of iterations of the algorithm, where the selection of a neighbour, evaluation and possible adoption of a new parameter set is a single iteration. The results from these experiments are presented in Section V, where each experiment was repeated 10 times and results averaged.



(a)



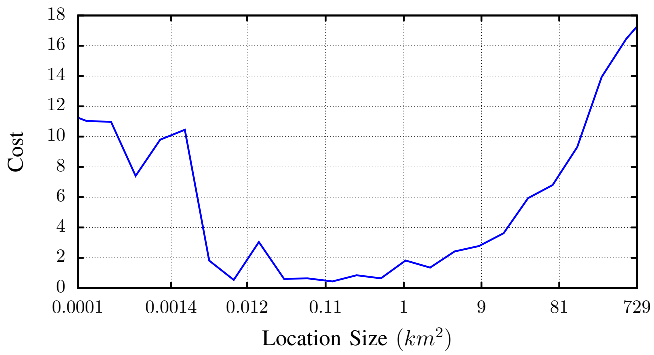
(b)

Fig. 1: Example simulated annealing runs showing MAE against time

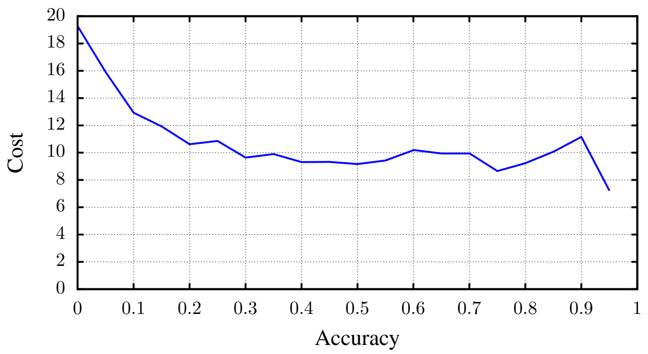
V. Results

Investigating the applicability of parameter optimisation to location extraction and prediction under the methodology proposed in Section IV, this section summarises the results obtained from the individual runs of the procedure. In total, 8,000 runs have been performed. Once a set of locations has been extracted and a prediction model trained, the resultant locations and predictions are given an associated cost by the MAE metric (discussed in Section III-A), where the optimal solution is the one with the lowest cost.

Figure 1 shows two example runs of the simulated annealing algorithm and how the cost of the extracted locations and predictions varies over time. In both cases, the final cost is significantly lower than the initial cost (where randomly selected parameters were used to extract locations and perform predictions). In Fig. 1b, the cost is monotonically decreasing and so each iteration produces a cost no worse than the the one before. Conversely, Fig. 1a shows steps that move to a position of higher cost on several occasions. This demonstrates simulated annealing's ability to overcome the local maxima



(a) Average location size



(b) Average predictive accuracy

Fig. 2: Relationship between MAE and extraction properties

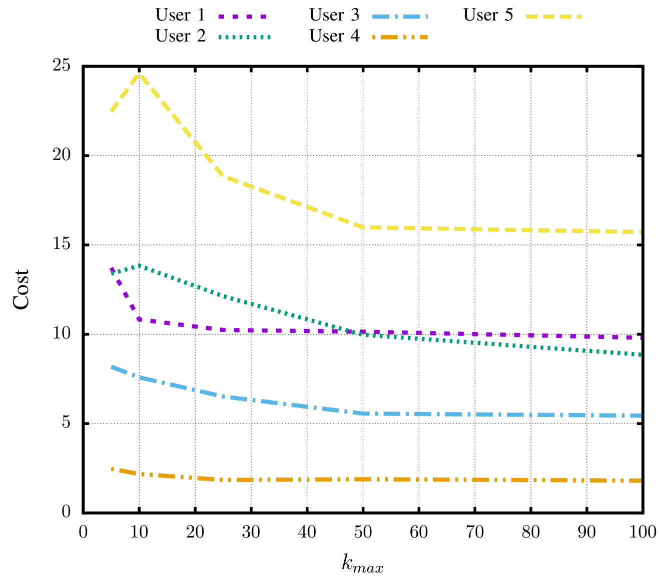


Fig. 3: Cost against number of iterations

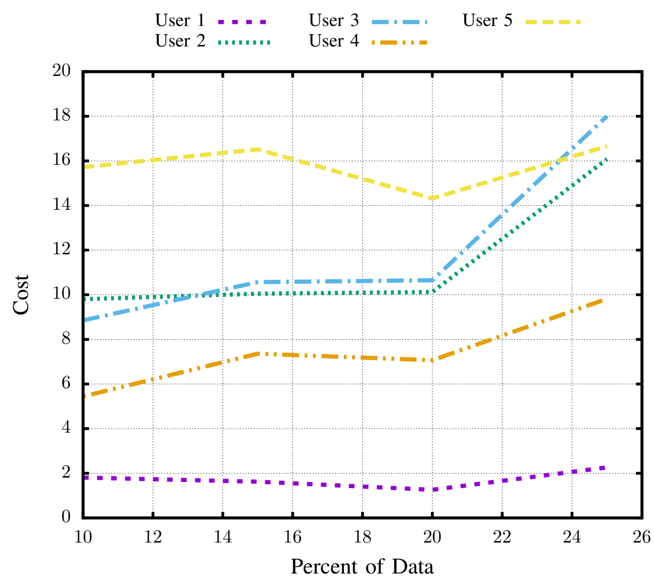


Fig. 4: Cost against percentage of data for different users

problem, taking worse positions towards the beginning of the run, but converging towards a minimum as time runs out. After $t = 43$, no move to a worse position is made, instead, the cost decreases before remaining constant.

Figure 2 shows the overall trends present with respect to the mean absolute error metric. Specifically, Figure 2a shows the relationship between cost and average location area (in km^2), and Figure 2b shows the relationship between cost and average predictive accuracy. Both graphs show the overall trends and show results from all iterations in all runs, demonstrating the link between these properties. Figure 2a shows that large locations incur a high cost, which decreases as the locations get smaller up to a certain point. Once locations become extremely small in size, they encompass few visits and thus provide less training data, leading to the cost increasing once again. Additionally, Figure 2b demonstrates that lower costs are indicative of higher predictive accuracy, up to a point. As predictive accuracy increases towards 1, however, the associated cost begins to increase. This is due to the situation discussed in Section III-A, where achieving such high predictive accuracy comes at the cost of location size. Due to the complexity of human mobility, achieving perfect predictions over small locations is extremely unlikely, and so in the cases where prediction accuracy approached 1, the locations became larger and thus were penalised with a higher score. Combined, these properties demonstrate that the definition of error as proposed in Section III-A favours a balance between small locations and high predictive accuracy. These are desirable properties for this purpose as they serve to accurately identify where a user will be in the future with as little uncertainty as possible.

Finally, Figures 3-5 show the effect on cost of the maximum number of iterations (i.e. k_{max}, Figure 3) and percentage of data used for each user (Figure 4 showing results when split per user, and Figure 5 when split by combination of visit extractor and predictor). In the case of Figure 3, the percentage of data used is held constant at 10%, and in the other two figures, k_{max} is held at 100, with results averaged over other properties (i.e. visit extractor, predictor and user). As evidenced by the figures, the metric performs as would be expected and as discussed in Section III-A. Specifically, as the number of iterations is increased, the average cost decreases slightly because the algorithm is allowed more

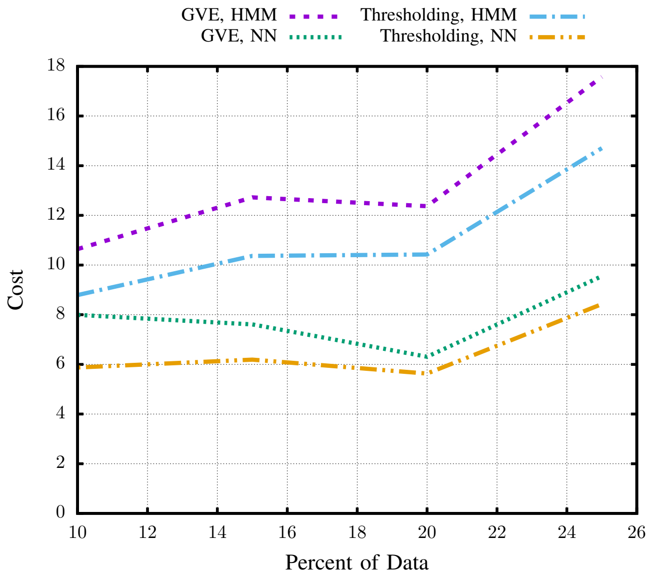


Fig. 5: Cost against percentage of data for different techniques

moves to find the optimal position (Fig. 3). While k_{max} can be further extended beyond 100, it was observed that with k_{max} of 100, 1,347 of 1,600 runs (i.e. 84%) had converged to a stable cost, indicating that the benefits of selecting a larger value for k_{max} would be minimal. Providing more of the user's data for optimisation results in increased costs, however (Figures 4 and 5). More data means that the predictor has more information to model the user's behaviour, but it also means that the user will likely have visited additional locations for which no previous transitions exist, increasing the complexity of the required predictive model and thus resulting in slightly increased costs as these new locations are likely to result in incorrect predictions. This combination of factors leads to the trends shown in the graphs, where the increase in information results in slightly higher average costs. Furthermore, Figure 5 shows similar trends across the different visit extraction and location prediction techniques employed, providing an indication of the generality of the MAE metric.

VI. Conclusion

This work has presented a method of automatic parameter optimisation for location extraction and prediction that understands the aims of both tasks. While existing work has assumed the validity of extracted locations and focused on prediction, we argue that predictions are predicated upon these locations and thus ensuring the representativness of such locations is of paramount importance to producing useful predictions.

Through a novel definition of error for location prediction, that considers the aims of the prediction process, combined with the simulated annealing mathematical optimisation algorithm, we provide a method of parameter optimisation that assures both the applicability of extracted locations and the utility of predictions. This applicability is then demonstrated through an evaluation of properties of the proposed metric. Such results evidence the conformity of the metric to the stated aim of minimising the distance between predicted future location and actual location visited by the user, thus identifying the region to be visited with as little uncertainty as possible.

The specific contributions of this paper have been: (i) the provison of a metric for the evaluation of both extracted locations and predictions that characterises the goal of each of these tasks, (ii) framing of the process of parameter selection as that of mathematical optimisation through the presented metric, and (iii) a discussion of characteristics of the metric along with a demonstration of its applicability over real-world data, location extraction algorithms and prediction techniques.

Focus for expanding the work presented in this paper should include evaluating the applicability of this approach on a larger range of algorithms, including additional location extraction techniques, prediction techniques and parameter optimisation algorithms, in conjunction with a greater emphasis on validation through unseen data.

VII. Acknowledgements

Our experiments used the MDC Database from Idiap Research Institute, Switzerland and owned by Nokia [16, 17].

References

[1] Sherif Akoush and Ahmed Sameh. Bayesian Learning of Neural Networks for Mobile User Position Prediction. In *Proceedings of the 12th International Euro-Par Conference on Parallel Processing*, pages 1234–1239, Honolulu, 2007.

[2] Gennady Andrienko, Natalia Andrienko, Georg Fuchs, Ana-Maria Olteanu Raimond, Juergen Symanzik, and Cezary Ziemlicki. Extracting Semantics of Individual Places from Movement Data by Analyzing Temporal Patterns of Visits. In *Proceedings of the ACM SIGSPATIAL International Conference on Advances in Geographic Information Systems*, pages 5–8, Orlando, 2013.

[3] Daniel Ashbrook and Thad Starner. Learning Significant Locations and Predicting User Movement with GPS. In *Proceedings of the 6th International Symposium on Wearable Computers*, pages 101–108, Seattle, 2002.

[4] Daniel Ashbrook and Thad Starner. Using GPS to Learn Significant Locations and Predict Movement Across Multiple Users. *Personal and Ubiquitous Computing*, 7(5):275–286, 2003.

[5] Roland Assam and Thomas Seidl. BodyGuards: A Clairvoyant Location Predictor Using Frequent Neighbors and Markov Model. In *Proceedings of the 10th IEEE International Conference on Ubiquitous Intelligence and Computing*, pages 25–32, Vietri sul Mere, 2013.

[6] Roland Assam and Thomas Seidl. Context-based Location Clustering and Prediction using Conditional Random Fields. In *Proceedings of the 13th International Conference on Mobile and Ubiquitous Multimedia*, pages 1–10, 2014.

[7] Athanasios Bamis and Andreas Savvides. Lightweight Extraction of Frequent Spatio-Temporal Activities from GPS Traces. In *Proceedings of the 31st Real-Time Systems Symposium*, pages 281–291, San Diego, 2010.

[8] Dimitris Bertsimas and John Tsitsiklis. Simulated Annealing. *Statistical Science*, 8(1):10–15, 1993.

[9] Pradeep Bilurkar, Narasimha Rao, Gowri Krishna, and Ravi Jain. Application of Neural Network Techniques

for Location Predication in Mobile Networking. In *Proceedings of the 9th International Conference on Neural Information Processing*, pages 2157–2161, Whistler, 2002.

[10] Yohan Chon, Hyojeong Shin, Elmurod Talipov, and Hojung Cha. Evaluating Mobility Models for Temporal Prediction with High-Granularity Mobility Data. In *Proceedings of the IEEE International Conference on Pervasive Computing and Communications Workshops*, pages 206–212, Lugano, 2012.

[11] Deborah F Cook, Cliff T Ragsdale, and Raymond L Major. Combining a neural network with a genetic algorithm for process parameter optimization. *Engineering Applications of Artificial Intelligence*, 13(4):391–396, 2000.

[12] Martin Ester, Hans-Peter Kriegel, Jörg Sander, and Xiaowei Xu. A Density-Based Algorithm for Discovering Clusters in Large Spatial Databases with Noise. In *Proceedings of the 16th International Conference on Knowledge Discovery and Data Mining*, pages 226–231, Portland, 1996.

[13] Holger Frohlich and Andreas Zell. Efficient Parameter Selection for Support Vector Machines in Classification and Regression via Model-based Global Optimization. In *Proceedings of the IEEE International Joint Conference on Neural Networks*, pages 1431–1436, Montreal, 2005.

[14] Yu Gong, Yong Li, Depeng Jin, Li Su, and Lieguang Zeng. A Location Prediction Scheme Based on Social Correlation. In *Proceedings of the IEEE Vehicular Technology Conference*, pages 1–5, 2011.

[15] Ramaswamy Hariharan and Kentaro Toyama. Project Lachesis: Parsing and Modeling Location Histories. In *Proceedings of the 3rd International Conference on Geographic Information Science*, pages 106–124, Adelphi, 2004.

[16] Niko Kiukkonen, Jan Blom, Olivier Dousse, Daniel Gatica-Perez, and Juha Laurila. Towards Rich Mobile Phone Datasets: Lausanne Data Collection Campaign. In *Proceedings of the First Workshop on Modeling and Retrieval of Context*, Berlin, 2010.

[17] Juha K Laurila, Daniel Gatica-Perez, Imad Aad, Jan Blom, Olivier Bornet, Trinh Minh Tri Do, Olivier Dousse, Julien Eberle, and Markus Miettinen. The Mobile Data Challenge: Big Data for Mobile Computing Research. In *Proceedings of the Nokia Mobile Data Challenge (MDC) Workshop in Conjunction with Pervasive*, Newcastle, 2012.

[18] Quannan Li, Yu Zheng, Xing Xie, Yukun Chen, Wenyu Liu, and Wei-Ying Ma. Mining User Similarity Based on Location History. In *Proceedings of the 16th ACM SIGSPATIAL International Conference on Advances in Geographic Information Systems*, pages 34:1–34:10, Irvine, 2008.

[19] James MacQueen. Some Methods for Classification and Analysis of Multivariate Observations. In *Procedings of the 5th Berkeley Symposium on Math, Statistics, and Probability*, pages 281–297, Berkeley, 1967.

[20] Raul Montoliu and Daniel Gatica-Perez. Discovering Human Places of Interest from Multimodal Mobile Phone Data. In *Proceedings of the 13th International Conference on Mobile and Ubiquitous Multimedia*, pages 12:1–12:10, Limassol, 2010.

[21] Pablo Moscato. On Evolution, Search, Optimization, Genetic Algorithms and Martial Arts: Towards Memetic Algorithms. *Caltech Concurrent Computation Program*, pages 1–67, 1989.

[22] Jan Petzold, Faruk Bagci, Wolfgang Trumler, and Theo Ungerer. Comparison of Different Methods for Next Location Prediction. In *Proceedings of the 12th International Euro-Par Conference on Parallel Processing*, pages 909–918, Dresden, 2006.

[23] Ginés Rubio, Héctor Pomares, Ignacio Rojas, and Luis Javier Herrera. A heuristic method for parameter selection in LS-SVM: Application to time series prediction. *International Journal of Forecasting*, 27(3):1–15, 2010.

[24] Stuart Russell and Peter Norvig. *Artificial Intelligence: A Modern Approach*. Prentice Hall, 4th edition, 2009.

[25] Lalit M Saini, Sanjeev K Aggarwal, and Ashwani Kumar. Parameter optimisation using genetic algorithm for support vector machine-based price-forecasting model in National electricity market. *Generation, Transmission & Distribution*, 4(1):36–14, 2010.

[26] Alasdair Thomason, Nathan Griffiths, and Matthew Leeke. Extracting Meaningful User Locations from Temporally Annotated Geospatial Data. In *Internet of Things: IoT Infrastructures*, volume 151 of *LNICST*, pages 84–90. Springer, 2015.

[27] Lucian Vintan, Arpad Gellert, Jan Petzold, and Theo Ungerer. Person Movement Prediction Using Neural Networks. In *Proceedings of the First Workshop on Modeling and Retrieval of Context*, Ulm, 2004.

[28] Jingjing Wang and Bhaskar Prabhala. Periodicity Based Next Place Prediction. In *Proceedings of the Nokia Mobile Data Challenge (MDC) Workshop in Conjunction with Pervasive*, Newcastle, 2012.

[29] Darrell Whitley, Timothy Starkweather, and Chris Bogart. Genetic algorithms and neural networks: optimizing connections and connectivity. *Parallel Computing*, 14(3):347–361, 1990.

[30] Vincent W Zheng, Yu Zheng, Xing Xie, and Qiang Yang. Collaborative Location and Activity Recommendations with GPS History Data. In *Proceedings of the 19th International Conference on World Wide Web*, pages 1029–1038, Raleigh, 2010.

[31] Yu Zheng, Lizhu Zhang, Xing Xie, and Wei-Ying Ma. Mining Interesting Locations and Travel Sequences from GPS Trajectories. In *Proceedings of the 18th International Conference on World Wide Web*, pages 791–800, Madrid, 2009.

[32] Jun Zhou, Qinpei Zhao, and Hongyu Li. Integrating Time Stamps into Discovering the Places of Interest. In *Proceedings of the 10th International Conference on Intelligent Computing Methodologies*, pages 571–580, Taiyuan, 2014.

[33] Matthias Zimmermann and Horst Bunke. Hidden Markov Model Length Optimization for Handwriting Recognition Systems. In *Proceedings of the 8th International Workshop on Frontiers in Handwriting Recognition*, pages 369–374, Niagara on the Lake, 2002.