

Advanced Location Prediction Techniques in Mobile Computing

Theodoros Anagnostopoulos*

National and Kapodistrian University of Athens
Department of Informatics and Telecommunications,
thanag@di.uoa.gr

Abstract— Context-awareness is viewed as one of the most important aspects in the emerging pervasive computing paradigm. Mobile context-aware applications are required to sense and react to changing environment conditions. Such applications, usually, need to recognize, classify and predict context in order to act efficiently, beforehand, for the benefit of the user. Firstly, we propose an efficient spatial context classifier and a short-term predictor for the future location of a mobile user in cellular networks. Secondly, we propose a novel adaptive mobility prediction algorithm, which deals with location context representation and trajectory prediction of moving users. Thirdly, we propose a short-memory adaptive location predictor that realizes mobility prediction in the absence of extensive historical mobility information. Fourthly, we assume the existence of a pattern base and try to compare the movement pattern of a certain user with stored information in order to predict future locations. Our findings, compared with other schemes, are very promising for the location prediction problem and the adoption of proactive context-aware applications and services.

1. Introduction

In order to render mobile context-aware applications intelligent enough to support users everywhere and anytime, information on the present *context* of the users has to be captured and processed accordingly. A well-known definition of context is the following: “*context is any information that can be used to characterize the situation of an entity. An entity is a person, place or object that is considered relevant to the integration between a user and an application, including the user and the application themselves*” [1]. Context refers to the current values of specific ingredients that represent the activity of an entity, situation and environmental state e.g., location, time, walking, attendance of a meeting, driving a car, traveling. One of the more intuitive capabilities of the mobile applications is their *proactivity*. The prediction of the user’s mobility behavior enables a new class of location-aware applications to be developed along with the improvement of the existing location-based services [2]. Even at the network level, mobility prediction assists in critical operations like handoff management, resource allocation, and quality of service provisioning.

Two classes of location (path) prediction schemes can be found in the current, mobile computing, literature. The first class includes schemes based on extensive historical data of the user movement. Such a scheme can be characterized as stateful. In a stateful scheme the prediction process relies on the *matching* of established (historical) movement patterns with the user movement experienced up to moment of prediction in order to estimate the future location of the user. Pattern-based and data mining approaches as well as machine learning techniques (e.g., learning automata) can be classified in this category. Contrary to the stateful scheme, a stateless model does not take into account extensive historical movement information for the prediction process. Instead, it uses of a *short sliding window* of historical movement information. Such scheme applies statistical techniques (e.g., extrapolation) on the recent movement information (window) in order to predict the future user location. The stateless scheme does not assume regularity in the user movement, as opposed to the stateful scheme, but proceeds with predictions based only on short-term spatiotemporal knowledge. Moreover, the prediction process comes along with adaptation techniques for certain parameters of the statistical techniques to fully cover the potential randomness of the user movement. One could also define hybrid schemes based on the stateful and stateless mechanisms that are invoked and collectively taken into account for joint decisions (e.g., weighted decisions).

*Dissertation Advisor: Stathes Hadjiefthymiades, Assistant Professor

2. Dissertation Summary

Firstly, we propose an efficient spatial context classifier and a short-term predictor for the future location of a mobile user in cellular networks [3], [4] and [5]. The process of location / context prediction is based on pattern classification in the sense that the user's mobility patterns are collected, processed, classified and used for predicting future mobility behavior. Typically, Machine Learning (ML) is primarily concerned with the design and development of algorithms for pattern classification and prediction. Such algorithms train a system (classifier) to classify observations in order to predict (predictor) unknown situations based on a history of patterns. Hence, if a mobile application is capable of tracing, learning and classifying the mobility patterns of a user, (i.e., the observed trajectories) then, it can predict his/her future mobility behavior. In this approach, we adopt pattern classification in order to predict the future location of a mobile user based on the spatial context i.e.,

- the current position and direction of the user,
- the history of the trajectories followed by the user, and
- the information on the user's surroundings (neighbouring network cells).

Each observed trajectory (pattern) is assigned to a specific class (a symbolic location) from a fixed set of classes. Our objective is to predict the symbolic location of a trajectory based on a collected set of popular trajectories. We propose a Location Predictor (LP), which is based on a Trajectory Classifier (TC), in order to address the location prediction problem. We design, implement and evaluate different versions of the proposed LP. Each variant exploits differently the derived knowledge on the mobility behavior of the user. The proposed, classifier-based, LP relies on short-term historical data of the user movement. Prediction is accomplished through the matching of established (historical) movement patterns with the short-term history of the user movement (spatiotemporal knowledge) seen up to the moment of prediction. In our case, a user mobility pattern is a series of symbolic locations. The size of the mobility pattern is directly linked to prediction accuracy. Limiting the size of the user's short-term history leads to loss of information, which results in false predictions and, on the other hand, redundant or excess historical knowledge results to noisy information and biased predictions. The challenge is to introduce an approach based on (a) short term spatio-temporal knowledge on user mobility behavior, and, (b) low computational cost on the decision predictor based on short-term historical movement data. The outcome of the proposed LP is provided through efficient and low complexity ML algorithms. Our findings show a clear advantage on the adoption of ML algorithms for location prediction as we obtain increased prediction accuracy (probability of correctness at a reasonable computing cost. In addition, we perform a comparative analysis of the proposed LP with popular prediction algorithms reported in the literature.

Secondly, we propose a novel adaptive mobility prediction algorithm, which deals with location context representation and trajectory prediction of moving users [6], [7]. In this context ML provides algorithms for learning a system to:

- clusters the user movements,
- identifies changes in the user movements
- adapts its knowledge structure to such changes, and,
- predicts the future user location.

In addition, ML provides solutions suitable for location prediction. Context-aware applications have a set of pivotal requirements e.g., flexibility and adaptation. These requirements would strongly benefit if the learning and prediction processes could be performed in real time. We argue that the most appropriate solutions for location prediction are offline and online clustering and classification. Offline clustering is performed through the Offline k Means algorithm. Online clustering is accomplished through the Online k Means and Adaptive Resonance Theory (ART). Offline learners typically perform complete model building, which can be very costly, if the amount of samples rises. On the other hand, online learning algorithms are able to (i) detect changes in patterns and (ii) update only parts of the model. The later provides adaptation of the model. Both forms of algorithms extract a subset of patterns, i.e., clusters in a knowledge base, from an initial dataset i.e., a database of user trajectories.

Online learning is more suited for the task of prediction through classification than offline learning. That is because, in real life, user movement data often needs to be processed in an online manner, each time after a new portion of the data arrives. This is caused by the fact that such data is organized in the form of a data stream (e.g., a sequence of time-stamped visited locations).

Classification involves the matching of an unseen pattern with existing clusters in the knowledge base. We rely on a

Hausdorff-like distance for matching unseen trajectories to clusters. Such metric applies to convex patterns and is considered ideal for user trajectories. Hence, location prediction boils down to location classification w.r.t. Hausdorff-like distance. In this approach, we introduce two training methods for training our algorithm: (i) the “nearly” *zero-knowledge* and (ii) the *supervised* method. In the first method, our algorithm is incrementally trained starting with a little knowledge on the user mobility behavior. In the second method, sets of known trajectories are fed to our algorithm. In addition, we introduce two learning methods for the proposed algorithm regarding the success of location prediction: (i) the *non-reinforcement learning* (nRL) and (ii) the *reinforcement learning* (RL). In the nRL method, a misclassified trajectory is no further used in the model-training phase. Hence, the algorithm is no longer aware of unsuccessful predictions. In the RL method, a misclassified trajectory is introduced into the knowledge base, thus, updating appropriately the model.

We evaluate the performance of our algorithm against real users’ movement. We establish important metrics for the performance assessment process w.r.t. (i) low system-requirements, i.e., storage capacity, and (ii) the required effort for model building, i.e., processing power. We assess the prediction accuracy of our algorithm, i.e., the precision of location predictions, along with the required size of the derived knowledge base. This size indicates the portion of the produced clusters out of the volume of the training patterns. Surely, we need to keep storage capacity as low as possible while maintaining good prediction accuracy. We also study the capability of our algorithm in adapting the derived model to unseen patterns. *Adaptivity* indicates the capability of the proposed algorithm to detect and update appropriately a specific part of the trained model. The algorithm should rapidly detect changes in the behavior of the mobile user and adapt accordingly through model updates. This is, however, achieved often at the expense of the prediction accuracy.

Thirdly, we propose a short-memory adaptive location predictor that realizes mobility prediction in the absence of extensive historical mobility information [8]. All the reported drawbacks led us to introduce a stateless scheme for movement prediction without considering user movement profiles and pattern-based techniques. The challenge is to introduce an approach based on

- short term spatiotemporal knowledge on user mobility behavior,
- detection and quick adaptation to changes of the user mobility behavior, and,
- very low requirements on storing and manipulating short-term historical movement data.

We present an approach for mobility prediction exploiting only the current spatiotemporal knowledge on the user’s mobility. Our aim is to address the problem of mobility prediction in the absence of large historical mobility information for the considered user(s). The proposed mobility prediction approach can be applied whenever the mobility behavior of a user is not known a-priori. For instance, consider tourists that visit a town or a freshman in a University campus. However, due to the absence of past mobility patterns, mobility prediction has also to regularly adapt to unseen user movements. The proposed scheme can, quite accurately, predict the traveling trajectory. This is achieved by using and dynamically adjusting only local current spatiotemporal knowledge on the user’s mobility behavior having very short relevant historical information. The main objective of our approach is to circumvent the difficulties that arise in predicting the user’s future location when extensive knowledge on the history of user’s traveling patterns is not available or the user behaves quite randomly. Another challenging point is the determination of the capacity of the relevant historical information used for mobility prediction. The capacity of the relevant historical information varies among mobile users since they exhibit different movement behaviors. In addition, more interestingly, the capacity of the relevant historical information is time-varying for a single user itself. This is because the mobile user can abruptly change his/her movement behavior, which has to be reflected in the capacity of the relevant historical information in order for the mobility prediction system to adjust to such unseen, new changes.

In this approach, we discuss the design and implementation of an *adaptive, short-memory* Location Predictor (LP). The LP does not require an extensive knowledge base of past user movements. Instead, the LP estimates the future location of the mobile user based on short-term movement information. That is, the LP exploits only local spatiotemporal knowledge of the user movement trying to predict future locations near in time. Evidently, this requires that the LP dynamically adjusts its model for future decisions on location estimations. Since there are no stored historical patterns or other representative knowledge on the user’s mobility behavior, the LP has also to be efficient even in the case of random (spontaneous) movement changes, e.g., sudden turns. On the other hand, the LP should reach a stable condition (model) whenever the user does not change his/her mobility behavior for long. Therefore, the LP should be able to rapidly detect changes in mobility behavior and adapt its model to the *current* behavior, thus, providing for fast adaptation of the model.

The criteria that we establish for the performance assessment of the adaptive LP take into account the system requirements (storage capacity) and the computational effort for prediction and fast adaptation. Besides the prediction accuracy, i.e., the precision of location predictions, we are interested in the size / capacity of the information that LP has to process for making predictions. We should stress that in our case, the consumed memory is extremely low if compared with other adaptive techniques for location prediction. Lastly, our objective is to assess the adaptive behavior of the LP that is its capability to rapidly detect changes in the movement of the mobile user and react accordingly.

Fourthly, we assume the existence of a pattern base and try to compare the movement pattern of a certain user with stored information in order to predict future locations [9]. Movement traces are always quite noisy and, therefore, undermine the performance of mobile computing mechanisms. A location-aware application requires online motion pattern learning and adaptation. On-line learning is more suited for the task of prediction. That is because, in real life, user movement data often needs to be processed in an on-line manner, each time new location information becomes available. However, most location-aware applications require training on the movement space of a moving object (supervised learning). Once the training is completed, the application is ready for prediction, and no additional modification is permitted. This case is acceptable provided that the motion patterns of the moving object correspond to well-defined boundaries. Nevertheless, in many realistic situations, the full spectrum of mobility patterns is not available during the training phase. The training process restricts the capability of the system to adapt its knowledge base to unseen patterns. The proposed system delays decision making (i) for trajectory classification and knowledge base adaptation, and for (ii) unseen pattern insertion in the knowledge base. Such delay makes the system 'confident' to decision making for classification, prediction, learning and adaptation. This system does not require a training process, learns, and revises the knowledge base in an on-line manner. Potential applications, as those mentioned above, which require efficient location prediction could adopt such delay tolerant decision making mechanism. Such functionality might enhance the prediction score for such location-aware applications by being more intelligent in decision making for location prediction and on-line motion pattern learning.

Our system manages the entire life-cycle of an on-line motion pattern learning, trajectory (sequential) classification, and adaptive movement prediction. It works in a relatively stateless way and tries to keep the underlying knowledge base K with the lowest possible total spatial variance if the stored patterns. The proposed system demonstrates some temporal tolerance (i.e., patience - delay) in classifying a trajectory to an existing (stored) pattern or abandon the similarity matching process and treat the assessed trajectory as completely new. It is worth noting that the proposed system is not forced to proceed with prediction once not being able to classify an unseen pattern. The system delays decision making for being more certain on prediction and learning decisions. We also introduce a trajectory similarity metric, which provides a holistic view of how close a stored pattern and an observed trajectory are. Should the classifier conclude, spatial variance reduction process decides (based on spatial statistics) on the exact pattern to be kept in the underlying knowledge base, K . If the new (assessed) trajectory demonstrates smaller spatial variance than the best matching pattern then the latter is substituted by the former. Otherwise K remains unmodified. In this approach we propose,

- a closed-loop system for sequential trajectory classification and k -step ahead location prediction;
- a dynamic programming approach (optimal stopping rules) for sequential trajectory classification;
- a trajectory similarity metric which supports sequential classification;
- an incremental and adaptive motion pattern learning scheme. The system does not require training phase and incrementally refines the knowledge base;
- a total spatial variance reduction process for the stored patterns in K , thus, avoiding classification decisions based on noisy patterns.

3. Results and Discussion

In the third approach, we adopt a local regression model based on kernel weighted functions in order to determine the future user location through extrapolation. The LP exploits a fuzzy controller in order to decide on the appropriate size of the memory of the local regression model that minimizes the prediction error. This means that, the fuzzy controller adjusts the memory size of the regression model based on the current user mobility behavior. Any detection of change on the mobility behavior is treated through a fuzzy control signal that adjusts the current memory size (history window) of the regression process. The overall model of the proposed LP is illustrated in Figure 1. Specifically, the regression model accumulates the last $m \geq 1$ locations and constructs a statistic regression function f at time instance t . Then, the future

location x , which is predicted for the next $l > 1$ time instances (i.e., at time index $t + l$), is an extrapolation point based on the mean value of the user velocity and direction of the last m locations. The prediction error e at time $t + l$ is calculated whenever the predicted location x is not the actual location at time $t + l$. Specifically the prediction error e is the spatial distance between the predicted location and the actual location at time $t + l$. The fuzzy controller is fed with the prediction error e and adjusts the length of the m last locations in the LP.

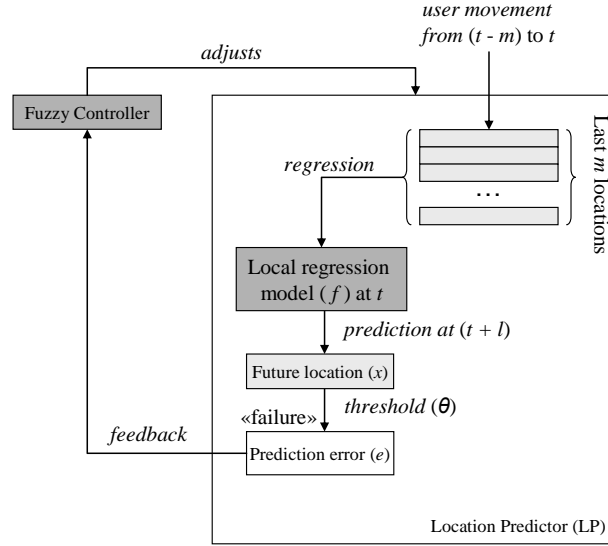


Figure 1. The adaptive short-memory Location Predictor (LP) model and fuzzy controller

3.1 Local Linear Regression Model

We now elaborate on the local regression component of the LP as illustrated in Figure 1. The estimating local regression function (model) $f(X)$ over a domain $X \subset \mathbb{R}^p$ fits a function f separately at each target point x_0 . This is achieved by using only those observations (points) close to x_0 to fit the model in a way that the resulting estimated function \hat{f} is *smooth* (has derivatives of all orders) in the domain \mathbb{R}^p . Local linear regression becomes less useful for $p > 2$. In our case $p = 1$ as discussed below. This localization is based on a weighted function or kernel $K_\lambda(x_0, x)$. The kernel assigns weights to x_i based on its distance from x_0 . The typical smoothing parameter λ indicates the width of the neighborhood. A larger λ implies lower variance. In such methods the model is structured by a set of observations and requires no training, i.e., all the work is performed at the location estimation time. The locally weighted regression solves a separate weighted least squares problem at each target point x_0 for a set of m observations $(x_i, f(x_i) = y_i)$. It should be noted that although we fit an entire linear model to the data in the region of m observations, we only use it to evaluate the fit at x_0 . The extrapolated $\hat{f}(x_0)$ is generated by the m previous observations $f(x_i)$, $i = 1, \dots, m$. The kernel function K_λ might be, for instance, the Epanechnikov quadratic kernel, defined as:

$$K_\lambda(x_0, x_i) = D\left(\frac{|x - x_0|}{\lambda}\right), D(u) = \begin{cases} \frac{3}{4}(1 - u^2), & |u| \leq 1 \\ 0, & \text{elsewhere} \end{cases}$$

According to such function $K_\lambda(x_0, x_i)$, as we “move” the point x_0 from left to right, neighboring points x_i enter the history window initially with weight zero, and then their contribution slowly increases. The regression model is used for interpolation among the location information of the user trajectory.

3.2 Location and Trajectory Information Model

We adopt a location and trajectory information model, in which the location of the user at time t is represented by a 2D vector $(x(t), y(t))$ of longitude $x(t)$ and latitude $y(t)$. The exact position information can be easily estimated. In our

case we use position information from GPS receivers embedded in the mobile user's handheld terminal. Based on the discussed representation, the history of the user movement up to time index n is represented as a sequence –trajectory- of n points $k(t) = (x(t), y(t))$, $t = 1 \dots n$, that is $\mathbf{k} = [k(t)] = [(x(1), y(1)), \dots, (x(n), y(n))]$. The sequence \mathbf{k} for $t = 1 \dots n$ is the user trajectory for the last n points starting from the point at time index $t = 1$. Each point $k(t)$ is sampled with frequency q , e.g., in our traces, the GPS locations are samples with $q = 1\text{Hz}$. A window \mathbf{w} is a sub-sequence of the m first points from \mathbf{k} , that is $\mathbf{w} = [k(t)]$, $t = 1, \dots, m$, $m < n$. The estimation-prediction of the future position of a mobile user is based on the window \mathbf{w} of m points. In other words, the predictor is fed with a trajectory of m points in order to predict future points. The discussed predictor is referred to as an $m+l$ predictor model [12]. That is the predictor estimates the future point indexed as $m+l$, i.e., predicts the point $k(m+l) = (x(m+l), y(m+l))$, with $l > 0$ from a trajectory (history window) of m points (see also Figure 2). For instance, we set $l=1$ for the prediction of the future point at the next time instance. The window of length m , the number of the l time indices for predicting the $k(l)$ point and the frequency of sampling q are the basic parameters in our LP, as illustrated in Figure 2.

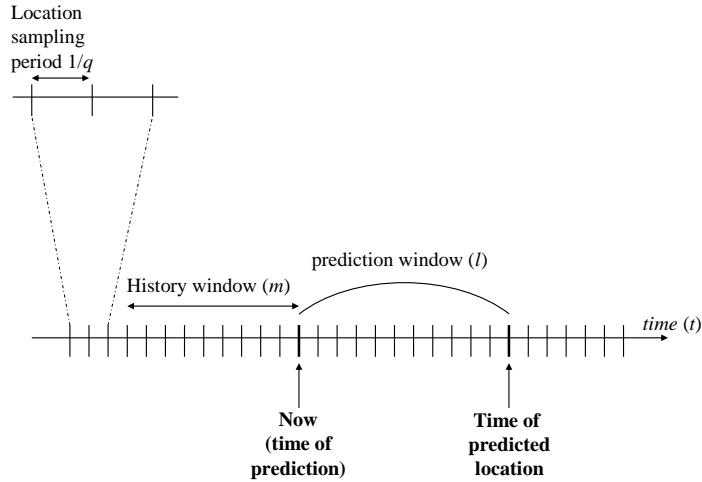


Figure 2. The basic parameters of the $m+l$ prediction.

3.3 Location Predictor Design

We discuss the adoption of the local linear regression in location prediction. We can distinguish the following cases for the user movement for a very small duration $\Delta t > 0$:

- Case-1: the user moves more steadily in the longitude $x(t)$ direction than in the latitude direction. This means that, it may hold true that the difference $|x(t+\Delta t) - x(t)| \ll |y(t+\Delta t) - y(t)|$ in a Δt .
- Case-2: the user moves more steadily in the latitude $y(t)$ direction than in the longitude direction, that is, $|y(t+\Delta t) - y(t)| \ll |x(t+\Delta t) - x(t)|$, and,
- Case-3: the user moves in such a way that $|x(t+\Delta t) - x(t)| = |y(t+\Delta t) - y(t)|$ in Δt .

Consider that for a time frame m , we calculate the mean differences for both longitude and latitude directions. That is, consider the vectors:

$$\begin{aligned} \Delta x(m) &= \frac{1}{m-1} \sum_{i=2}^m (x(i) - x(i-1)) = \frac{1}{m-1} (x(m) - x(1)) \\ \Delta y(m) &= \frac{1}{m-1} \sum_{i=2}^m (y(i) - y(i-1)) = \frac{1}{m-1} (y(m) - y(1)) \end{aligned} \quad (1)$$

The predictor can estimate the point after l time instances when the following cases holds true:

- If $|\Delta x(m)| < |\Delta y(m)|$ then $k(m+l) = (x(m+l), y(m+l))$ with the extrapolation $x(m+l) = x(m) + l \cdot \Delta x(m)$ and the local regression estimate $y(m+l) = \hat{f}(x(m+l))$. In this case, the $y(m+l)$ is generated by the window \mathbf{w} and the local linear estimation of $\hat{f}(x(m+l))$ at $x(m+l)$ having local regression in the 1-dimension domain ($p = 1$);

- If $|\Delta x(m)| > |\Delta y(m)|$ then $k(m+l) = (x(m+l), y(m+l))$ with the extrapolation $y(m+l) = y(m) + l \cdot \Delta y(m)$ and the local regression estimate $x(m+l) = \hat{f}(y(m+l))$. In this case, the $x(m+l)$ is generated by the window w and the local linear estimation of $\hat{f}(y(m+l))$ at $y(m+l)$.

It is worth noting to study the case where $|\Delta x(m)| = |\Delta y(m)|$. Specifically, the time window m used for the calculation of $|\Delta x(m)|$ or $|\Delta y(m)|$ may contain smaller spatial transitions not captured by the absolute values of relocation in the x or y axis. A small time window m renders the scheme more sensitive to such (small) transitions while a large time window may ‘absorb’ movements and derive predictions that are not representative of the actual movement (not necessarily wrong predictions). Surely, due to the above shortcoming, if $|\Delta x(m)| = |\Delta y(m)| = 0$, the predictor fails to distinguish between the totally static case and a recurring (cycle) path. Hence, in this case, the future point $k(m+l)$ is $(x(m+l), y(m+l))$ with the extrapolation either $y(m+l) = y(m) + l \cdot \Delta y(m)$ (and $x(m+l) = \hat{f}(y(m+l))$) or $x(m+l) = x(m) + l \cdot \Delta x(m)$ (and $y(m+l) = \hat{f}(x(m+l))$).

In order to judge whether the predictor is successful in predicting the future point $k(m+l) = (x(m+l), y(m+l))$, we introduce an error threshold $\theta > 0$. Such threshold is the minimum distance of the actual point $z(m+l)$ at time instance $m+l$ and the predicted point $k(m+l)$. The $z(m+l)$ point is the actual (observed) location of the mobile user at time instance $m+l$. Hence, a prediction is considered successful if $\|z(m+l) - k(m+l)\| \leq \theta$. Otherwise, the prediction fails.

If we use the $m+l$ predictor for N times (for large N) then we can estimate the *probability of a successful $m+l$ prediction* $P(m,l)$. $P(m,l)$ is the fraction of the correctly predicted positions, i.e., those predicted points $k(m+l)$ that fall within a circle of radius θ , out of N predictor invocations, i.e.,

$$P(m,l) = \frac{|\{k(m+l) : \|z(m+l) - k(m+l)\| \leq \theta\}|}{N}$$

where $z(m+l)$ is the actual (observed) location of the mobile user at time instance $m+l$ and $|U|$ is the cardinality of the set U .

3.3 Fuzzy Control for Adaptive Location Prediction

Fuzzy control is a direct nonlinear mapping between inputs (e.g., location prediction error, change of the prediction error) and outputs (e.g., change in the memory size m). Fuzzy control provides formal techniques to represent and implement human experts’ heuristic knowledge for controlling a system. Such techniques rely on certain *if-then* rules instead of relying on mathematical modeling of the system. Fuzzy logic controllers are used extensively, especially, in the form of the proportional, integral and derivative (PID), PD, and PI controllers. In this paper we consider a fuzzy, proportional-integral controller PI_m for adjusting the window length m in location prediction. The control signal is based on a set of fuzzy rules that generate the required output.

Figure 2 depicts the high level structure of the proposed closed-loop, real-time system for adjusting the history window length m . To achieve the maximum value of the probability of successful predictions $P(m(t), l)$ at time t , which is denoted by the reference model $a(t) = 1$ at time t (the upper bound in the probability of successful predictions), the PI_m controller computes the required value of m , denoted as $u(t)$, based on the error $e(t)$, change of error $\Delta e(t)$ and the past decision of the controller on changing m , $\Delta m(t)$. Specifically, the error $e(t)$ at time instance t is:

$$e(t) = a(t) - P(m(t), l) \quad (2)$$

In location prediction, we aim at maximizing the $P(m(t), l)$ at any given t that is, $a(t) = 1$ at each prediction. The error change at time t is:

$$\Delta e(t) = e(t) - e(t-1) \quad (3)$$

and the change of the controller’s past decision on m is:

$$\Delta m(t) = m(t) - m(t-1) \quad (4)$$

The exact value of u which is the window length that minimizes the error cannot be directly estimated. Instead, the PI_m adjusts $m(t)$ in order to minimize $e(t)$ for each t . In addition, it should be noted that the m value that minimizes $e(t)$ (denoted by m^*) is not constant since the mobile user can spontaneously change its moving behavior. In such case, the controller has to adapt to such change and adjust m so as to minimize $e(t)$ until the next change of the user’s behavior.

We define the required change in the value of m , $\Delta u(t) \in [-1, 1]$, which denotes the signed portion of changing the value of $m(t)$ -increase, decrease or leave constant- and results to the value of $m(t + 1)$ at the next time step, $t+1$. In other words we obtain the following adaptation rule:

$$m(t+1) = m(t) + \mu(\Delta u(t)m(t)) \quad (5)$$

with $\mu > 0$ an adaptation (learning) rate. The proposed fuzzy controller calculates the value of the control signal $\Delta u(t) = u(t) - u(t - 1)$. That is, the decision on the next value of m (that minimizes the error) is based on the past decision on m along with the observed error induced by the LP. The change in m , Δm , and the change in error, Δe , are the basic components in our controller which, along with the value of error, determine the direction (positive, zero, negative) of Δu in the decision on the next value of m , as will be described subsequently.

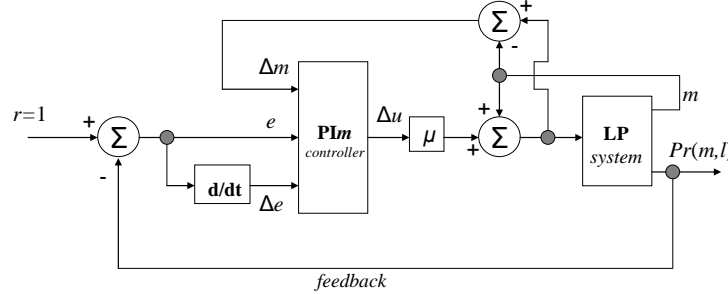


Figure 2. The control flow and feedback of the fuzzy controller PI_m to the location predictor LP

The recursive equation of a discrete PID controller is:

$$u(t) = G_p e(t) + G_I \sum_{\tau=0}^t e(\tau) + G_D \Delta e(t)$$

where G_p , G_I , G_D are the corresponding constant proportional, integral and derivative gains. The definition of the difference between two consecutive control outputs $\Delta u(t)$ is:

$$\Delta u(t) = u(t) - u(t-1) = G_I e(t) + G_p \Delta e(t) + G_D \Delta^2 e(t) \quad (6)$$

where $\Delta^2 e(t) = \Delta e(t) - \Delta e(t - 1)$. In our case, the PI_m has three inputs $-e(t)$, $\Delta e(t)$, and $\Delta m(t)$ - and one output -the current change in m , $\Delta u(t)$. Our PI_m controller has the following form:

$$u(t) = G_p e(t) + G_I \sum_{\tau=0}^t e(\tau) + G_D (\Delta e(t) + \Delta m(t)) \quad (7)$$

The iterative form of $u(t)$ is obtained by taking the derivative of both sides of the Equation (8) having also $\Delta m(t) = m(t) - m(t - 1)$, thus, in our case,

$$\Delta u(t) = G_p \Delta e(t) + G_I e(t) + G_D (\Delta^2 e(t) + \Delta^2 m(t)) \quad (8)$$

and then, $u(t) = u(t - 1) + G_I e(t) + G_p \Delta e(t) + G_D (\Delta^2 e(t) + \Delta^2 m(t))$ with $\Delta^2 m(t) = \Delta m(t) - \Delta m(t - 1)$. Based on Equation (8), the inputs of our controller (e , Δe and Δm) and the output ($\Delta u(t)$) turn the controller into a PI controller. Figure 2 depicts the architecture of the proposed controller.

3.5 Fuzzy Controller

We describe the basic fuzzy control system for inferring the required control signal based on fuzzy inference rules. A fuzzy controller is a fuzzy logic system with n inputs and k outputs. In our case, the PI_m controller is a Multiple Input Single Output (MISO) fuzzy logic system with $n = 3$ and $k = 1$, such that the input at time t is $\mathbf{p}(t) = [e(t), \Delta e(t), \Delta m(t)]$ and output $\mathbf{q}(t) = [\Delta u(t)]$. The fuzzy system consists of the processes:

- fuzzification,

- fuzzy inference process, and
- defuzzification.

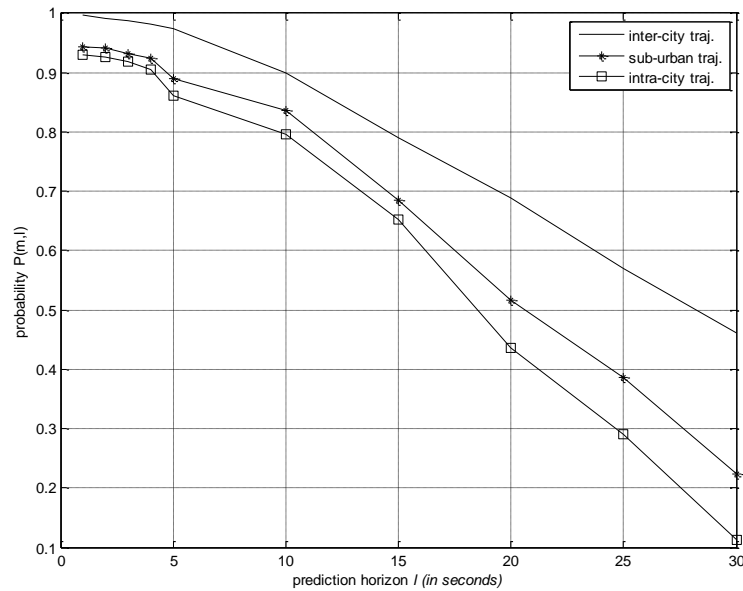


Figure 3. Probability of successful prediction for $m=10$ s and different l values

3.6 Performance of the Fuzzy Controlled Location Predictor

We assess the performance of the proposed LP and the capability of the proposed PI_m controller to control the LP targeting to adaptively minimize the prediction error. At first, we examine the performance of the LP by experimenting with real GPS traces. We examine the probability of correct predictions $P(m, l)$ and determine the best value of the window length m^* that minimizes the prediction error. The value m^* cannot be determined beforehand (e.g., analytically) due to the inherent randomness in the mobility behavior of humans. For this reason, we require that the proposed controller adjusts the value of m in order to converge to m^* . Once the user changes its mobility behavior, the PI_m has to re-adjust m . With the aim of evaluating the PI_m controller, we have determined the best values for m for which the LP assumes minimum prediction error. Hence, we examine whether the PI_m controller converges to such values. Moreover, we examine the adaptive behavior of the controller. That is, its capability to detect changes in the mobility behavior of the user and to re-adjust its decisions to new m^* values.

3.7 Experimental Movement Trajectories - Traces

We examine the behavior of the proposed LP with real movement traces of mobile users in German, Italy, France, and Denmark. In those traces the mobile user moves

- among different locations in a city (termed intra-city trajectory),
- among different suburban areas of a city (termed suburban trajectory), and,
- among different cities in highways (inter-city trajectory).

3.8 Performance Assessment of the Fuzzy Controlled Location Predictor

At first, we examine the probability of successful predictions derived from LP by experimenting with various values of m and l without introducing the PI_m controller. Figure 3 depicts the probability of successful prediction $P(10, l)$ for various values of l for $N = 500$ predictions. Such size of m is not the most suitable for making predictions with minimum

prediction errors. As illustrated in Figure 3, for a low value of l (short-term prediction), $P(10, l)$ assumes high values. For instance, for $l = 5 < m$, $P(10, l)$ assumes high values for any type of trajectories ($r = 0.5$). In addition, for $r = 1$, that is $m = l = 10$, the probability of prediction assumes values 0.8 and 0.9 for intra-city and intercity trajectories, respectively. When the value of l increases the prediction error also increases, for constant m . For $r = 2$, $P(10, 20)$ assumes values between 0.45 to 0.7 for all types of trajectories.

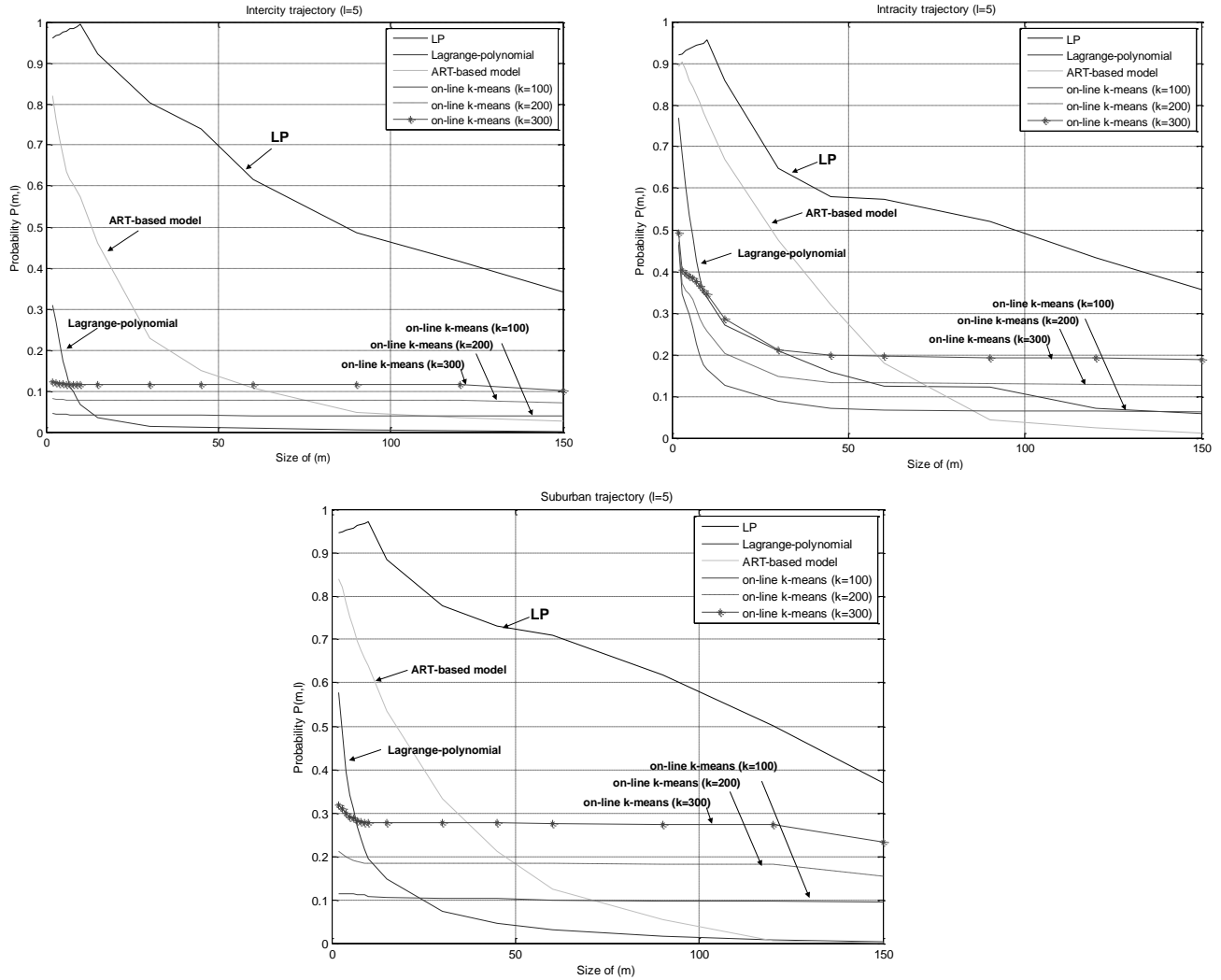


Figure 4. The $P(m, l)$ vs. size of m for the different trajectory types

3.9 Comparison with other Models

We compare the behavior of our stateless model with other stateful and stateless models w.r.t. the probability of successful prediction, space requirements and time complexity. At first, we compare our model with the (stateless) Lagrange polynomial method for location information extrapolation (prediction). In addition, we experiment with two stateful models, the Adaptive Resonance Theory (ART) model enhanced with reinforcement learning and the on-line k -means algorithm. Such models are based on mobility patterns and use pattern classification for a given trajectory in order to predict the future user location. It is worth noting that all considered models adapt their spatiotemporal knowledge base to the mobility behavior of the user. The main criterion for comparison is then how accurately and efficiently, in terms of space and time, each adaptive model predicts the future user location.

3.10 Comparative Performance Assessment

At first, we examine the probability of successful prediction $P(m, l)$ obtained by the stateless and stateful models for the three trajectory types. We experiment with the ‘ $m+l$ ’ prediction with $l = 5$ seconds, vigilance h and threshold value θ set to 10 meters, and $k = 100, 200$ and 300 in the on-line k -means algorithm. Figure 4 depicts the $P(m, l)$ (vs. the size m) achieved by the algorithms for the intercity, intra-city and sub-urban trajectories. We observe that the LP algorithm achieves better $P(m, l)$ values than the other models in all types of trajectories; specifically, Figure 4 depicts the m^* value for LP, in which $P(m^*, l)$ is maximum ($m^* = 10$ for $l = 5$). On the other hand the ART-based model achieves good prediction accuracy, especially, for values of $m < 25$ for the three trajectory types. Specifically, the ART-based model demonstrates good prediction accuracy for the intracity trajectory (only 10% lower than that of LP) due to the fact that the mobile user ‘repeats’ some mobility patterns moving within the city. However, for $m > 50$ the ART-based model assumes very low $P(m, l)$ and for $m > 100$ it shows the worst performance. Furthermore the Lagrange polynomial model obtains quite good prediction accuracy when m is low. Specifically, it obtains $P(m, l) = 0.78$ with $m = 5$ for the intercity trajectory. For $m > 10$ the Lagrange polynomial is not suitable for predictions. Finally all the on-line k -means models ($k = 100, 200$ and 300), for every m and in all kinds of trajectories, achieve low performance in terms of prediction accuracy. This is attributed to the fact that such model is not able to increase the predefined number of clusters. Instead it can only readjust them. It is worth noting that the on-line k -means model is independent of the value of m , except for the case of the intra-city trajectory, in which $m < 5$. But, even in this case the performance in prediction accuracy is low.

4. Conclusions

In the first approach we proposed efficient LP schemes based on ML algorithms for trajectory classification. Specifically, the proposed spatial context classifier and a short-term predictor for predicting the location of a mobile user in cellular networks exploits (i) the current position and direction of the user, (ii) history of the trajectories of the user, and, (iii) surrounding location information. We design, implement and evaluate different variants of the proposed LP. Each variant exploits differently the derived knowledge on the mobility behavior of the user (namely the macro and the micro LP). We define the parameters of the short-term LP and introduce certain metrics for evaluating the ability of correct predictions and the efficiency in the prediction process. Moreover, we compare our LP (all the corresponding variants) with popular predictors discussed in the relevant literature. Such predictors are also based on ML algorithms. Simulations with synthetic and real-world mobility data shown that the proposed short-term MiD predictor achieves high prediction efficiency and accuracy, thus, delivering LPs suitable for advanced context-aware applications.

In the second approach we presented how ML can be applied to the engineering of mobile context-aware applications for location prediction. Specifically, we proposed an adaptive ML algorithm for location prediction using ART (a special Neural Network Local Model). We introduce two learning methods: one with non-reinforcement learning and one with reinforcement learning. Furthermore, we deal with two training methods for each learning method: in the supervised method the model uses training data in order to make classification and in the zero-knowledge method the model incrementally learns from unsuccessful predictions. We evaluated our models (versions of the proposed algorithm) with different spatial and temporal parameters. We examine the knowledge bases storage cost (i.e., emerged clusters) and the precision measures (prediction accuracy). Our findings indicate that the C-RLnT suits better to context-aware systems. The advantage of C-RLnT is that (1) it does not require pre-existing knowledge in the user movement behavior in order to predict future movements, (2) it adapts its on-line knowledge base to unseen patterns and (3) it does not consumes much memory to store the emerged clusters. For this reason, C-RLnT is quite useful in context-aware applications where no prior knowledge about the user context is available. Furthermore, through experiments, we decide on which vigilance value achieves the appropriate precision w.r.t. memory limitations and prediction error.

In the third approach we study the proactivity feature of mobile, location-dependent applications and present an approach for mobility prediction exploiting only recent user movement knowledge. We propose a short-memory adaptive LP to address the problem of mobility prediction in the absence of extensive historical information. The location prediction of the proposed LP is obtained by a local linear regression model, while the adaptive capability is achieved through a fuzzy-driven PI_m controller. Such controller produces control signals for estimating the best size for the mobility history window attempting to minimize the location prediction error. We experiment with real GPS traces and examine the predictability and adaptability behavior of our LP.

The LP dynamically controls the size of the historical mobility behavior, by the produced control signals, for intra-city,

sub-urban and inter-city trajectories. LP stabilizes to small m^* when dealing with sudden movement changes. This leads to minimizing the probability of inducing noise into the local regression model. In addition, we can conclude that the control signals are more aggressive in the case of intra-city trajectory than in intercity trajectory since the former movement integrates abrupt movement changes while the latter does not. Finally, we show that LP achieves fast adaptation to sudden changes experienced when the user changes mobility behavior (i.e., transition between different trajectory types).

In the forth approach, we propose a sequential trajectory classification and spatial variance reduction system for noise resilient movement prediction of moving objects. The system deals with noisy motion patterns due to random deviations from previously seen patterns, which negatively impacts the accuracy of the prediction result. The system relies on stochastic dynamic programming which relaxes the classification task so that slightly different patterns can be treated as equivalent. Moreover, the system adopts SVRP for keeping K concise and with minimum spatial variance. We provided a comparative assessment of the model with on-line classifiers and a variant adopting the odds algorithm. The proposed model achieves high prediction scores along with efficient data storage of motion patterns.

References

- [1] A. Dey, Understanding and using context, *Personal and Ubiquitous Computing*, 5(1), pp. 4-7, 2001.
- [2] J. Hightower, G. Borriello, *Location Systems for Ubiquitous Computing*, IEEE Computer, 34(8), August, 2001.
- [3] T. Anagnostopoulos, C. Anagnostopoulos, S. Hadjiefthymiades, *Efficient Location Prediction in Mobile Cellular Networks*, International Journal of Wireless Information Networks, Springer, Springer Verlag, December, 2011 [doi:10.1007/s10776-011-0166-9].
- [4] T. Anagnostopoulos, Christos Anagnostopoulos, Stathes Hadjiefthymiades, M. Kyriakakos, A. Kalousis, *Predicting the location of mobile users: a machine learning approach*, ACM International Conference on Pervasive Services (ICPS09), Imperial College, London, UK., July, 2009.
- [5] T. Anagnostopoulos, C. Anagnostopoulos, S. Hadjiefthymiades, A. Kalousis, M. Kyriakakos, *Path Prediction through Data Mining*, IEEE Conference on Pervasive Services, ICPS07, Istanbul, Turkey, July, 2007.
- [6] T. Anagnostopoulos, C. Anagnostopoulos, S. Hadjiefthymiades, *An Adaptive Machine Learning Algorithm for Location Prediction*, International Journal of Wireless Information Networks (2011) 18:88-99, Springer, April, 2011.
- [7] T. Anagnostopoulos, C. Anagnostopoulos, S. Hadjiefthymiades, *An Online Adaptive Model for Location Prediction*, (3rd Conf.) ICST Autonomics 2009, Limassol, Cyprus, September, 2009.
- [8] T. Anagnostopoulos, C. Anagnostopoulos, S. Hadjiefthymiades, *An Adaptive Location Prediction Model based on Fuzzy Control*, Computer Communications 34 (2011) 816-834, Elsevier, Elsevier, January, 2011.
- [9] C. Anagnostopoulos, S. Hadjiefthymiades, T. Anagnostopoulos, *Intelligent Trajectory Classification for Improved Movement Prediction*, IEEE Transactions on Mobile Computing, November, 2011., [revision].