UNIVERSITY OF TARTU
Institute of Computer Science
Computer Science Curriculum

Tanel Kiis

Stop Detection and Location Accuracy Improvement in Mobile Positioning

Master's Thesis (30 ECTS)

Supervisor: Toivo Vajakas, MSc

Tartu 2018

Stop Detection and Location Accuracy Improvement in Mobile Positioning

Abstract:

Mobile operators collect data about their clients' activity in the mobile network. Each event made in the mobile network has reference to the antenna the mobile device was connected to at that time. By knowing the coverage areas of the antennas the peoples' trajectories throughout the day can be approximated. Its spatial coarseness and temporal sparseness makes extracting information from this data a compelling task requiring specially crafted algorithmic tools. Detecting when and where did the mobile device stopped is a crucial step that serves as a basis for subsequent data analysis tasks on this data. Here a state of the art stop detection algorithm is analysed and some shortcomings of it identified. The proposed improvements to these have been shown to improve the performance of the stop detection algorithm significantly. Additionally, the possibility of improving the location accuracy by incorporating periodicity into the movement models is investigated.

Keywords:
Mobile positioning, stop detection, Kalman filtering, Guassian process regression

CERCS:
P170 - Computer science, numerical analysis, systems, control

Peatuste tuvastamine ja mobiilpositsioneerimise asukohatäpsuse parandamine

Lühikokkuvõte:

Mobiilioperaatorid koguvad andmeid oma klientide tegevuse kohta mobiilivõrgus. Igal võrgusündmusel on viide mastile, mille külge oli seade parasjagu ühendatud. Teades mastide kattealasid, saab hinnata inimeste trajektoore läbi terve päeva. Nende trajektooride halva ruumilise lahutuse ja ajalise hõreduse tõttu on neist kasuliku informatsiooni eraldamine väljakutsuv ülesanne, mis vajab spetsiifilisi algoritmilisi lahendusi. Trajekotooridest paigalseisude tuvastamine on aluseks mitmetele küsimustele lahenduse leidmisele. Selles töös on uuritud ühte paigalseisude leidmise algoritmi ning on tuvastatud mõned selle puudujäägid. Nende puuduste parandamisel muutus algoritm märksa täpsemaks ja töökindlamaks. Lisaks on uuritud trajektooride perioodilisuse kaasamise mõju mudelite asukohahinnangutele.

Võtmesõnad:
Mobiilpositsioneerimine, paigaleisude tuvastamine, Kalmani filter, Gaussi protsessid

CERCS:
P170 - Arvutiteadus, arvutusmeetodid, süsteemid, juhtimine (automaatjuhtimisteooria)

Contents

1 Introduction

1.1 Motivation

Mobile operators collect data about their clients' activity in the mobile network. This data can be used internally to improve the quality of the service by optimizing the mobile network. Several techniques have been designed to automatically detect faults in the network [1, 2]. Researchers have also taken interest in the possibilities of this data - it contains a very large portion of the total population. The location information in it can be used to perform different large-scale mobility analysis [3, 4]. There has also been a lot of research into applications that potentially hold significant business value: road usage analysis [5], smart city design [6] and even identifying potential clients for businesses [7].

All these applications rely on the accuracy of the location data and the ability to reliably extract information from it. One of the most fundamental information that can be extracted is when and where did the mobile device stop. With this knowledge a wide range of questions can be answered, such as "Where do they live," "Which establishments are they visiting," "Were they held up in a traffic jam?"

1.2 Structure of the Thesis

In the introductory chapter, the field of mobile positioning data analysis is introduced and its appeal to researchers and businesses is described. A brief overview of the research done in this field is given and finally, the contributions of this work are listed. Chapter 2 focuses on improving a state of art stop detection algorithm. In chapter 3 a method to incorporate periodic mobility patterns into trajectory modelling is proposed. In chapter 4 the developed algorithms are applied to real data and their performances are evaluated. Chapter 5 summarizes the work done in this thesis. In the appendix there is the list of used acronyms and the licence.

1.3 Background

The overview of the field of mobile positioning is largely based on the work of T. Vajakas [8].

1.3.1 Network Events

The data collected by the mobile operator about its clients is a timestamped series of events made in the network. These events can be caused by users activity on the phone or they can be initialized by the network. Each mobile positioning data record has an attribute identifying the mobile network cell the mobile device was connected to, known as the Cell Global Identity (CGI). A cell is a geographical area where it is possible to connect to the transceiver of a base station. The operators design

smaller cells in regions of high population density, so cell would service approximately the same number of devices. This makes the location information in urban areas significantly more accurate. The neighbouring cells have a considerable overlap making it possible to connect to several cells from any location [8].

Mobile operators have approximated the shapes of all the cells in their network. The real cell shape depends on many factors, such as antenna radiation pattern and height, network load [9], signal attenuation on the landscape [10], signal reflections and radio interference [11], network configuration parameters such as handover threshold and neighbour cell lists [12]. All the models used in this work operate under Gaussian uncertainty assumption, hence the cell has to be approximated by a binomial Gaussian probability distribution. Throughout this work, this is done such that the 1σ radius covers the cell.

1.3.2 Related Work

Extracting mobility patterns from mobile data can face a lot of noise due the "ping-pong handover" phenomena. This is due to the device being handed over between antennas when an overload of the network occurs. This phenomenon creates a sort of noise in the trajectories or mobility episodes. Thus, several techniques have been developed to reduce "ping-pong" distortions [8, 13–15].

When it comes to mobility episode there is a interesting research focusing on handling noisy trajectories in stops detection. Fiadino et al presented a study on trajectory reconstruction, where they used a "ping-pong" suppression method that ignores events where the device connects back to the previous cell within a predefined time window [13]. In [14] the authors describe a method for computing edit distances between event sequences where short-term handovers to another cell and repeated events can be ignored.

For the mobile network data Calabrese et al [15] used a method that was inspired by earlier work for GPS data in [16] and [17], which performs a clustering of measurement points and replaces original events with the barycenter of the cluster. Clustering has been used also for example in form of sequence analysis providing very good computational throughput [8] or requiring more computations with density-based clustering [18].

Another approach that dealt with detecting the mobility episodes was proposed in [19], where the authors presented a technique based on Switching Kalman filter (SKF) for differentiating between the movement episodes - Stay, Jump and Move. This method is further analysed in chapter 2.

1.4 Contributions of This Work

The main goal of this work is improving the algorithms that are used for mobile positioning data analysis. The first contribution of this work is analysing the Switching Kalman filtering algorithm proposed by Batrashev et al [19]. Three

shortcomings of that algorithm are identified and solutions to overcome them are proposed. The effectiveness of the improvements is evaluated on a real data.

The second contribution is designing a method to incorporate periodic movement patterns into the trajectory models and evaluating the effectiveness of this for location accuracy improvement. For this method, some limitations have been detected, that require further investigation.

2 Methodology of Stop Detection

One state of the art approach to stop detection is a Switching Kalman filtering based algorithm proposed by Batrashev et al [19]. Although they achieved good results with their methods, there were some implicit assumptions made, that rarely hold for mobile network events. In this work three of such limitations have been identified and solutions proposed for them. Two of these assumptions were related to the time differences between consecutive events made by a device in the mobile network. The third limitation involved overconfident location accuracy for stationary devices, caused by correlated measurement errors.

Before addressing these limitations, a concise overview of Kalman filtering (KF) and Switching Kalman Filtering has to be given. In this section KF and SKF are introduced similarly how they were in the original publication by Batrashev et al and then the shortcomings are analysed and improvements proposed in the following subsections. In this section time is viewed as discrete time steps $t-1$, t, $t+1$, etc. Quantities that change through time have the time step t marked in the subscript.

2.1 Kalman and Switching Kalman Filtering

Kalman filter is a recursive algorithm for finding maximum likelihood estimations (MLE) for the hidden state of a linear dynamic system [20]. This is done by collecting observations that are related to the hidden state. All the observations are assumed to have a Gaussian white noise with known covariance. Formally there are two main equations, that define the dynamic liner system, upon which the Kalman filter will be applied:

$$\mathbf{x}_t = \mathbf{F}_t \mathbf{x}_{t-1} + \mathbf{q}_t \tag{1}$$

$$\mathbf{y}_t = \mathbf{H}_t \mathbf{x}_t + \mathbf{r}_t \tag{2}$$

Equation (1) describes the change of the hidden state $\mathbf{x}$ from time step $t-1$ to time step t. The matrix $\mathbf{F}_t$ is known as the state transition matrix and this can be different for each time t. Vector $\mathbf{q}_t$ is a Gaussian noise with covariance matrix $\mathbf{Q}_t$. Equation (2) relates the hidden state with the observations. $\mathbf{H}_t$ is the observation matrix, unlike $\mathbf{F}_t$ this does not have to be a square matrix. The Observation noise $\mathbf{r}_t$ has covariance $\mathbf{R}_t$ and $\mathbf{y}_t$ are the observations made.

The actual value of $\mathbf{x}_t$ will remain unknown and so the best prediction given observations up to time t' is used: $\hat{\mathbf{x}}_{t|t'}$. This prediction also has an error estimate, that can be described by a covariance matrix: $\mathbf{P}_{t|t'}$. Kalman filtering algorithm for finding these optimal estimates requires the following equations to be applied for every time step t:

$$\hat{\mathbf{x}}_{t|t-1} = \mathbf{F}_t \hat{\mathbf{x}}_{t-1|t-1} \quad \text{the noise is centered at zero} \tag{3}$$

$$\mathbf{P}_{t|t-1} = \mathbf{F}_t \mathbf{P}_{t-1|t-1} \mathbf{F}_t^T + \mathbf{Q}_t \tag{4}$$

$$\hat{\mathbf{e}}_t = \mathbf{y}_t - \mathbf{H}_t \hat{\mathbf{x}}_{t|t-1} \quad \text{measurement residual} \tag{5}$$

$$\mathbf{S}_t = \mathbf{H}_t \mathbf{P}_{t|t-1} \mathbf{H}_t^T + \mathbf{R}_t \quad \text{residual covariance} \tag{6}$$

$$\mathbf{K}_t = \mathbf{P}_{t|t-1} \mathbf{H}_t^T \mathbf{S}_t^{-1} \quad \text{optimal Kalman gain} \tag{7}$$

$$\hat{\mathbf{x}}_{t|t} = \hat{\mathbf{x}}_{\mathbf{t|t-1}} + \mathbf{K}_t \hat{\mathbf{e}}_t \quad \text{updated estimate} \tag{8}$$

$$\mathbf{P}_{t|t} = (\mathbf{I} - \mathbf{K}_t \mathbf{H}_t) \mathbf{P}_{t|t-1} \quad \text{updated estimated covariance} \tag{9}$$

Kalman filtering provides optimal estimation given only the past information. This is ideal for real-time applications, but it can be improved for offline uses. Fixed interval smoothing is used to obtain the MLE for the hidden state and its the covariance of errors, that uses the all the observations up to time $T > t$. The algorithm for this is named after its creators Rauch, Tung and Striebel (RTS):

$$\mathbf{C}_t = \mathbf{P}_{t|t} \mathbf{F}_{t+1}^T \mathbf{P}_{t+1|t}^{-1} \quad \text{RTS gain} \tag{10}$$

$$\hat{\mathbf{x}}_{t|T} = \hat{\mathbf{x}}_{t|t} + \mathbf{C}_t (\hat{\mathbf{x}}_{t+1|T} - \hat{\mathbf{x}}_{t+1|t}) \quad \text{smoothed state} \tag{11}$$

$$\mathbf{P}_{t|T} = \mathbf{P}_{t|t} + \mathbf{C}_t (\mathbf{P}_{t+1|T} - \mathbf{P}_{t+1|t}) \mathbf{C}_t^T \quad \text{smoothed state covariance} \tag{12}$$

These formulas are applied iteratively, starting from time T, as a result, the MLE for all the hidden states given all the observations are found [21].

If the nature of the underlying process changes in time, then this can be modelled by changing the matrices $\mathbf{F}_t$, $\mathbf{Q}_t$, $\mathbf{H}_t$ and $\mathbf{R}_t$ correspondingly. In the context of this work, the most important change in the process is whether the observed mobile device is moving or staying still. Classical Kalman Filter requires that this change is known in advance. To overcome this limitation an algorithm called Switching Kalman filter has been designed [22].

Switching Kalman filter requires a fixed set of Kalman filters to be chosen, such that each state transition could be described by a linear combination of these. All of these Kalman filters are applied to every time step t and the measurement residuals $\hat{\mathbf{e}}_t$ are used for finding the probabilities for each of the models that this transition occurred according to that model. In some cases these probabilities can be even more important results of the algorithm than the estimation of $\hat{\mathbf{x}}_{t|t}$, that is the weighted average of the predictions of all the models.

In this work three mobility types are distinguished and corresponding models associated with them: Stay, Jump and Move. The Stay model describes a stationary device. Jump and Move are both describing a non-stationary device: under the Jump model the device changes location without leaving a trail of events during the actual movement, the Move model, on the other hand, is describing the change

in location that is accompanied by a sequence of events during the movement. In the Jump model, the change in location is modelled by having a very large location uncertainty for a short period. The Move uses the speed of the device as the main cause for the location change.

For even more powerful estimation, a model switching matrix $\mathbf{Z}_t$ can be specified. An element of this matrix $\mathbf{Z}_{tij}$ is the probability that if the transition to time step $t-1$ was made according to the model i, then the transition to time step t will be done according to model j. The most common use of this matrix is forcing the consecutive transitions to be more likely made by the same model, meaning that the diagonal elements would be significantly larger than the off-diagonal elements. The only exception to this is the Jump model: several consecutive jumps should be covered by the Move model.

When modeling the movements of mobile devices, the hidden state vector has four components: $\mathbf{x}_t = \begin{pmatrix} x \\ y \\ \dot{x} \\ \dot{y} \end{pmatrix}$, two for location and two for speed. Observation vector on the other hand has only two components: $\mathbf{y}_t = \begin{pmatrix} x \\ y \end{pmatrix}$ [19].

The observation matrix is time-independent and same for all the models:

$$\mathbf{H}_t = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \end{pmatrix}.$$

For Stay and Jump models, the state transition matrix $\mathbf{F}_t$ is a unit matrix. State transition matrix of the Move model corresponds to simple linear movements:

$$\mathbf{F}_t = \begin{pmatrix} 1 & 0 & \Delta t & 0 \\ 0 & 1 & 0 & \Delta t \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}. \tag{13}$$

In this equation Δt is time between time steps $t-1$ and t. For each model, the covariance matrix of the observation noise corresponds to the size of the coverage area of the antenna, where the device was connected to.

The most significant difference between the models comes from their state transition noise covariance matrices $\mathbf{Q}_t$:

- The Stay model should have the smallest variance in both speed and location. The real hidden state does not change at all according to this model, but the prediction can vary when new observations are acquired.
- In the Move model, the uncertainty of location is also small, but speed is allowed to vary greatly to model more complex movements: turning, acceleration, etc.
- While Jump model includes a change in location, it is not caused by the speed, but by very high uncertainty in the location. The variance in speed is set to be as low as in the Stay model.

Finding optimal values for these covariances can be done with parameter optimization.

Similarly to the Kalman filter, the Switching Kalman filter also considers only the past observations when computing the maximum likelihood estimations. Fortunately, the SKF also has the possibility for a smoothing step. The set of equations for this is significantly more complex and is very well covered in the original article [22].

2.2 Proposed Improvements to Current Method

2.2.1 Sampling Rate Independent Model Switching

The time between two consecutive events in the mobile network can differ greatly between different devices and also throughout the day for any particular device. While actively using mobile internet, the network can register several events per minute, but when the device stays idly, there can be up to two hours between events and sometimes even more. An event density distribution for a real mobile network is depicted in Figure 1. Noticeable features on that distribution are the peaks at some times, for example, 42.5 min and 2 h. These can be caused by some internal procedures of the mobile network or by some devices that periodically provide some updates. Exact locations of these peaks can change for different mobile operators, but the general shape stays the same.

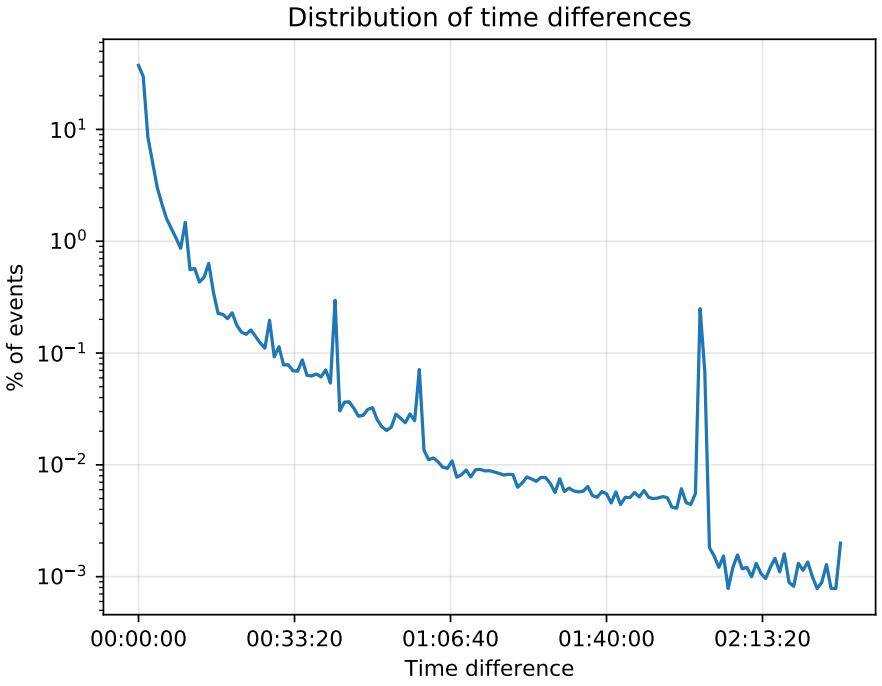

Figure 1. Distribution of time differences between consecutive successful network events for a mobile device. This distribution has very long and thin tail that is clipped from the plot.

In the overview of the Kalman and Switching Kalman filters, at two places an implicit

assumption was made that the time difference between two consecutive events is constant. In reality, it can change at least two orders of magnitude. First of these was made, when the model switching matrix $\mathbf{Z}_t$ was chosen to be fixed. In the article it was proposed to use the following values:

$$\mathbf{Z}_t = \begin{pmatrix} 0.8 & 0.1 & 0.1 \\ 0.1 & 0.8 & 0.1 \\ 0.45 & 0.45 & 0.1 \end{pmatrix}.$$

The ordering of the models is Stay, Move, Jump. This sort of matrix defines a discrete Markov chain (MC) between these three states [23]. To illustrate how the change in the frequency of the events changes the model, a toy example can be considered. A person is standing still at time $t = 0$, its change in mobility type is described by a Markov chain with $\mathbf{Z}_t$ as its transition matrix. When disregarding the observations, then its probability to be staying still after time t has passed is shown in Figure 2. It is paradoxical, that the persons behaviour should change depending on the number of events it does in the mobile network. The use of fixed probabilities Z_t causes devices, that make more events, have a higher chance to change its movement model without having any extra evidence for it.

When using the given values for $\mathbf{Z}_t$, then the asymptotic value of the probability for staying still is 45 %. This is independent of the sampling rate and should express the probability of staying still when there have been no observations from this device in a long time. Mobile devices are most commonly carried by people with them, so it is reasonable to assume that the mobility patterns of mobile devices are rather similar to those of people. Analysis of peoples mobility in urban areas has shown, that on average every person takes 2.8 trips every day, with the average length of 26 minutes [24, 25]. This is about 5 % of the day on the move - 11 times less than the 55 % proposed by the values in $\mathbf{Z}_t$.

One method for calculating the values of $\mathbf{Z}_t$ in such way that the sampling rate has no effect on the outcome is using a continuous-time Markov chain (CTMC). Unlike discrete Markov chain, that is parametrized with transition probabilities, the continuous-time MC is parametrized with transition rates [23]. The non-negative transition rates λ_{ij} express the number of transitions that take place on average in a unit of time from state i to state j. Transitions from state i to i are not modelled. These values are arranged in a generator matrix $\mathbf{A}$, such that for off-diagonal elements $\mathbf{A}_{ij} = \lambda_{ij}$ and the diagonal elements are chosen such that all rows add up to 0. By solving either the Kolmogorov forward or backward equations, an expression for $\mathbf{Z}_t$ is obtained

$$\mathbf{Z}_t(\Delta t) = e^{\mathbf{A}\Delta t},$$

Δt is the time between time step t and $t - 1$ [23]. Matrix exponentiations is done using the eigenvalue decomposition $\mathbf{A} = \mathbf{B}\mathbf{\Lambda}\mathbf{B}^{-1}$ and Taylor expansion. Resulting in the following equation:

$$\mathbf{Z}_t(\Delta t) = \mathbf{B}e^{\mathbf{\Lambda}\Delta t}\mathbf{B}^{-1}.$$

$\mathbf{\Lambda}$ is a diagonal matrix and the result of exponentiation of a diagonal matrix is another diagonal matrix whose elements are elementwise exponents of the initial matrix. The computational cost of this negligible compared to the rest of the algorithm.

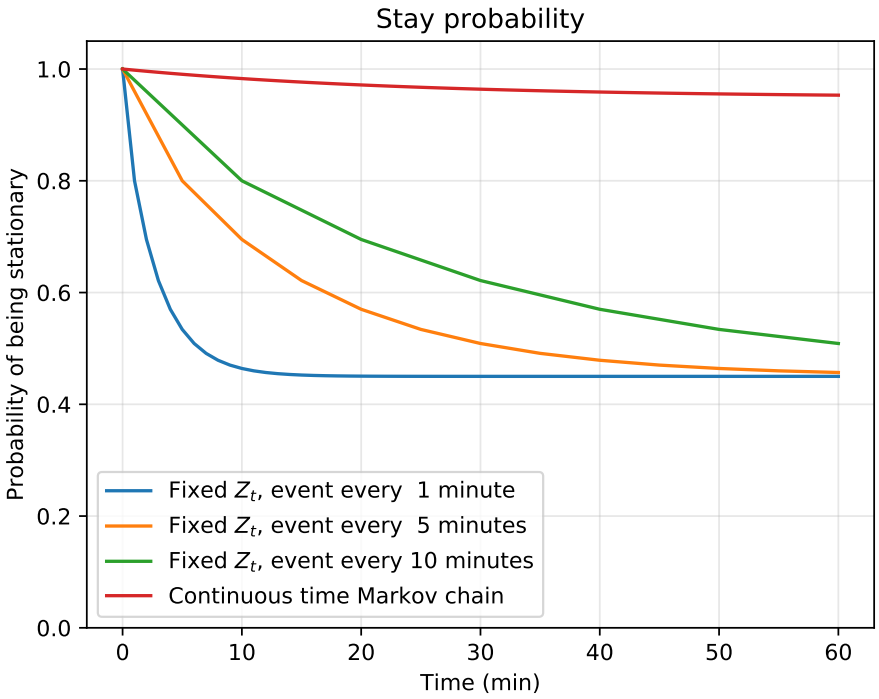

Figure 2. The probability that a stationary person is stationary after time t depending on the choice of $\mathbf{Z}_t$ and sampling rate. It is important to note that the person does not have to be stationary for the entire duration..

The need for separate Move and Jump models does not come from peoples mobility patterns, but from the behaviour of the mobile network. Because of this, statistical papers cannot be used to differentiate these models. The mobile network data shows that about 10 % of trips are happening according to the jump model and others are taking place according to the Move model. When also considering the average length of a trip and the average number of trips in a day, the generator matrix can be estimated:

$$\mathbf{A} = \begin{pmatrix} -0.125 & 0.113 & 0.012 \\ 2.31 & -2.31 & 0 \\ 2.31 & 0 & -2.31 \end{pmatrix} \frac{\text{transitions}}{\text{h}}. \tag{14}$$

The ordering of the models is Stay, Move, Jump. As a comparison, the probability that a stationary person, who behaves according to the generator matrix $\mathbf{A}$, is stationary after time t is added to the Figure 2.

2.2.2 Sampling Rate Independent Process Noise

In the original article, the values for process noise covariance $\mathbf{Q}_t$ were fixed [19]. This simplification assumes that the time difference Δt between time steps is constant. As seen in the previous subsection, this assumption does not hold. To get a better understanding about the effect this assumption has, a simple example can be considered: a person with known initial location and speed ($x(0) = 0\,\text{m}$, $\dot{x}(0) = 1\,\frac{\text{m}}{\text{s}}$) is observed

over a period of 1 h. Its movement is modelled as linear ($\mathbf{F} = \begin{pmatrix} 1 & \Delta t \\ 0 & 1 \end{pmatrix}$) with fixed process noise $\mathbf{Q} = \begin{pmatrix} 0.01 & 0 \\ 0 & 0.01 \end{pmatrix}$. Its predicted location ($\hat{x}$) and its variance are depicted in Figure 3. The predicted location is precisely the same for any sampling rate. The confidence in this prediction of the other hand decreases as sampling rate increases. This implies that while the state transition matrix $\mathbf{F}_t$ is correct, the process noise covariance $\mathbf{Q}_t$ should depend on Δt somehow.

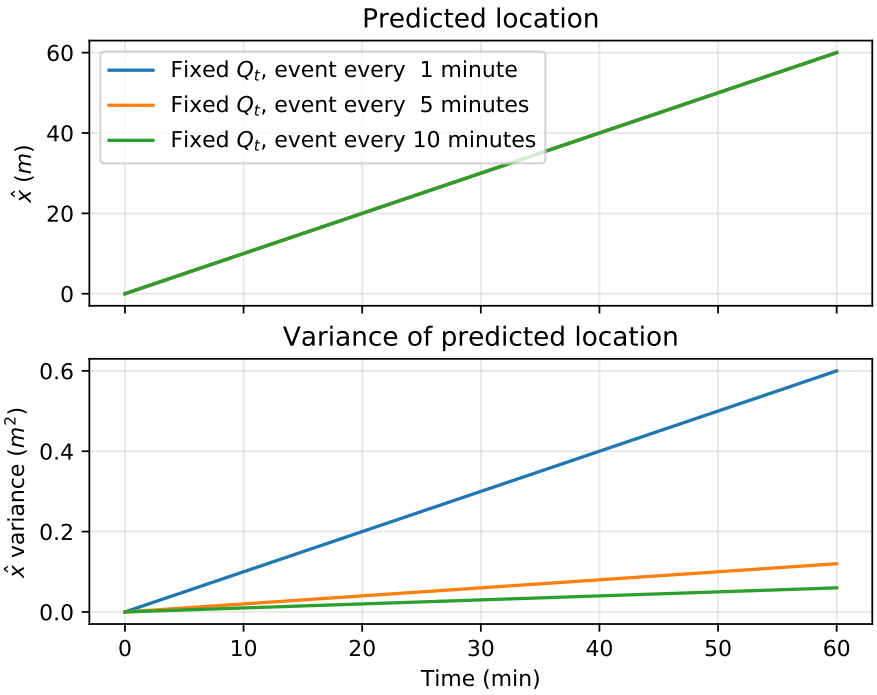

Figure 3. Predicted location and its variance for a person, whose initial position and velocity are perfectly known. All three lines on the $\hat{x}$ plot overlap perfectly.

The following derivation of $\mathbf{Q}_t(\Delta t)$ is based on the article of P. Axelsson and F. Gustafsson [26]. The state transition equation (1) is a discretized version of stochastic differential equation

$$\mathbf{d}\mathbf{x}(t) = \mathbf{A}\mathbf{x}(t)dt + \mathbf{d}\beta(t), \tag{15}$$

where $\beta(t)$ is a Brownian motion with

$$E[\mathbf{d}\beta(t)\mathbf{d}\beta(t)^T] = \mathbf{S}dt. \tag{16}$$

Discretization is done by integrating the equation (15) over time interval $[t_{t-1}, t_t]$. The integration results is

$$\underbrace{\mathbf{x}(t_t)}_{\mathbf{x}_t} = \underbrace{e^{\mathbf{A}\Delta t}}_{F_t} \underbrace{\mathbf{x}(t_{t-1})}_{\mathbf{x}_{t-1}} + \underbrace{\int_{t_{t-1}}^{t_t} e^{\mathbf{A}(\tau - t_t)} \mathbf{d}\beta(\tau)}_{q_t}. \tag{17}$$

Using the definition of covariance of the white noise $E[q_{t_1} q_{t_2}] = \mathbf{Q}_{t_1} \delta_{t_1 t_2}$, the covariance matrix $\mathbf{Q}_t$ can be expressed as

$$\mathbf{Q}_t = \int_0^{\Delta t} e^{\mathbf{A}\tau} \mathbf{S} e^{\mathbf{A}^T \tau} d\tau. \tag{18}$$

Solving this sort of integral for a general case is a rather complicated problem, but there are some approaches to it - for example using Lyapunov equations. When all of the eigenvalues for matrix $\mathbf{A}$ are zeros, then this integral will have an analytical solution. This condition holds for all three of the models: Stay, Move and Jump. For Move model the matrix $\mathbf{A}$ is

$$\mathbf{A} = \begin{pmatrix} 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{pmatrix},$$

for Stay and Jump models, it is zero matrix. The solution to integral (18) under the assumption that all the eigenvalues of $\mathbf{A}$ are zeros is

$$\mathbf{Q}_t = \sum_{i=0}^{p-1} \sum_{j=0}^{p-1} \frac{\Delta t^{1+i+j}}{i! j! (1+i+j)} \mathbf{A}^i \mathbf{S} \mathbf{A}^{jT},$$

where p is the dimensionality of the square matrix $\mathbf{A}$. Because of the sparsity of $\mathbf{A}$ in these models, most of the the terms are 0, resulting in simplified expression for $\mathbf{Q}_t$, for Stay and Jump model it will be

$$\mathbf{Q}_t = \mathbf{S} \Delta t \tag{19}$$

and for move Model it will be

$$\mathbf{Q}_t = \mathbf{S} \Delta t + (\mathbf{A}\mathbf{S} + \mathbf{S}\mathbf{A}^T) \frac{\Delta t^2}{2} + \mathbf{A}\mathbf{S}\mathbf{A}^T \frac{\Delta t^3}{3}. \tag{20}$$

There is no reason to believe that the noise for different components of $\mathbf{x}$ would be directly correlated, so matrix $\mathbf{S}$ can be a diagonal matrix. The values on diagonal can satisfy the same conditions that were proposed for a fixed $\mathbf{Q}$. Selection of these values by parameter optimization is discussed in the results section.

2.2.3 Correlation and Overconfidence

One of the preconditions for applying Kalman Filter is that the process noise $\mathbf{q}_t$ and the observation noise $\mathbf{r}_t$ have to be uncorrelated in time [20]. There is no reason to believe that this assumption does not hold for the process noise, but there is one case where it does not hold for the observation noise. When the device is stationary, it is rather common, that it will keep connecting to the same cell over a long period. When considering each of these connections as an uncorrelated measurement, then the predicted hidden state will converge to the observation, as depicted in Figure 4.

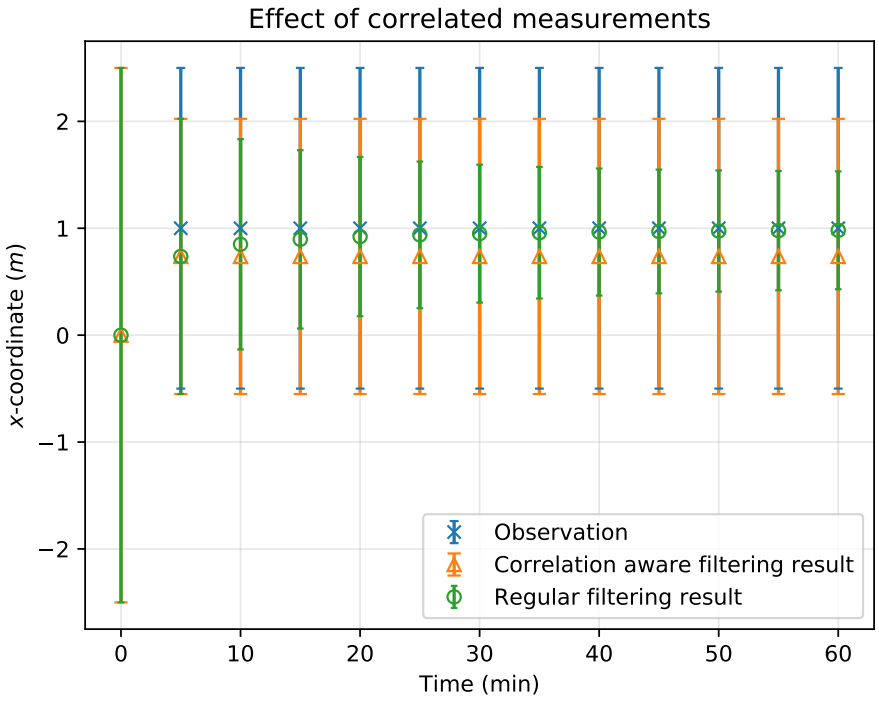

Figure 4. Behaviour of Kalman filters predicted location and covariance with correlated measurements. At time 0 the prior probabilities are shown.

This results in unrealistically overconfident predictions. Completely ignoring these consecutive events for Stay model would also be unwary, because even when fully correlated, they might be still carrying information with them.

There has been some work done to design a Kalman filter generalization for correlated noise. Here the results of Petovello et al [27] are considered. In their work, they generalized observation equation (2) to also include correlated error:

$$\mathbf{y}_t = \mathbf{H}_t\mathbf{x}_t + \mathbf{u}_t. \tag{21}$$

In this new relation, $\mathbf{u}_t$ is time-correlated error, with correlation expressed through equation

$$\mathbf{u}_t = \mathbf{\Psi}_t\mathbf{u_{t-1}} + \mathbf{r}_t. \tag{22}$$

As in the original Kalman filter, the error $\mathbf{r}_t$ is a white Gaussian noise with covariance $\mathbf{R}_t$. A concise overview of filtering a dynamic system, that follows this relation has been given by Wang et al, along with a comparison to alternative approaches [28].

As stated before, the correlation does not effect the Jump or Move models. The improved Stay model will have two possible cases:

- If the observation is different from the previous, then under this model there is no correlation and the matrix $\mathbf{\Psi}_t$ is zero. In this case, the regular Kalman filter can be applied.
- If observation $\mathbf{y}_t$ is the same as the previous one $\mathbf{y}_{t-1}$, then under this model the correlation transition matrix $\mathbf{\Psi}_t$ is a unit matrix and there is no added

white noise, meaning that $\mathbf{R}_t$ is zero.

When substituting these assumptions along with the state transition matrix for Stay model $\mathbf{F}_t = \mathbf{I}$ into the new update formulas presented in [28], then they will simplify significantly:

$$\hat{\mathbf{x}}_{t|t} = \hat{\mathbf{x}}_{t-1|t-1}, \tag{23}$$

$$\mathbf{P}_{t|t} = \mathbf{P}_{t-1|t-1}. \tag{24}$$

According to this result, correlated measurements have no effect on the estimated hidden state $\hat{\mathbf{x}}$, but they will compensate the effects of the process noise on the covariance of the prediction $\mathbf{P}$. This can be taught as a stationary device, that stays connected to the same cell, is suspended in time according to the observer, until it will connect to a new cell. An example how this kind of model behaves under correlated measurements is depicted in Figure 4. As expected, the correlation aware model does not converge to an overconfident prediction but stays at a realistic value.

When using a Switching Kalman filter with regular Move and Jump models along with correlation aware Stay model, then there will still be some convergence, because the final result is a weighted average of these three models. This is not a problem to be solved, but a sign of the predictive power of SKF. When there is strong evidence, that the device is stationary, then the weight of stay model will be very close to 1 and the convergence is negligible.

The RTS smoothing algorithm also requires changes for it to work with correlated measurements. Just like in the filtering case, this is only applied to the Stay model. If observation $\mathbf{y}_t$ is not the same as the next one $\mathbf{y}_{t+1}$, then regular RTS smoothing equations can be used. Otherwise, the correlation aware update equations must be used:

$$\hat{\mathbf{x}}_{t|T} = \hat{\mathbf{x}}_{t+1|T}, \tag{25}$$

$$\mathbf{P}_{t|T} = \mathbf{P}_{t+1|T}. \tag{26}$$

3 Incorporating Periodicity into Trajectory Models

Modelling trajectories as series of points is a reasonable simplification, that still retains a significant amount of its predictive power. In this work, it has provided predictions of movement types and also predicted the location of the mobile device with improved accuracy. There are still questions about trajectories, that can not be answered with the results obtained so far. A common task, that arises when doing trajectory analysis is to give a prediction of the location of the subject at an arbitrary time t, that might not have a measurement associated with it. This prediction must contain both the most likely location and a covariance matrix of the prediction error. Some sort of regression model would be appropriate for this task.

A seemingly unrelated question to this is how can more domain knowledge about peoples movements be included in the models, to improve their predictive power. It has been shown that peoples trajectories contain a lot of periodic patterns [3]. Using these periodic patterns could give significantly more accurate location predictions for periods with very sparse measurements.

One possible regression model, that has a natural way to deal with periodic functions is Gaussian process regression (GPR). In this section, an overview of GPR is given and its viability as a tool for a trajectory analysis assessed. As an improvement over Gaussian process regression, a GPR mixture model (GPRMM) is introduced and an alternative to the expectation maximisation algorithm of GPRMM is proposed. GPR will be used as a second layer upon the outputs of the switching Kalman filter.

3.1 Gaussian Process Regression

The theoretical overview of Gaussian process regression is based on the book of Rasmussen and Williams [29]. An intuitive definition for Gaussian processes proposed by them is "a Gaussian process is a collection of random variables, any finite number of which have a joint Gaussian distribution". Gaussian processes can be used for both regression and classification, but only regression is covered in this work.

3.1.1 General Overview

Gaussian process is completely specified by two functions: mean $m(t)$ and covariance $k(t, t')$. Although in this work $m(t)$ is considered to be constant for a trajectory, equalling to the geometric centroid of the trajectory, in the following formulation of GPR a general mean function is used. Trajectories have two coordinates, because of this there have to be two separate means: one for easting ($m_E(t)$) and other for northing ($m_N(t)$). It is shown that peoples trajectories tend to be very anisotropic, meaning there have to be also separate covariance functions for easting ($k_E(t, t')$) and northing ($k_N(t, t')$) [3]. To avoid modelling the covariance between easting and northing at any time the trajectory is transformed into its principal components,

that are uncorrelated with each other [30]. These components will still be referenced to as easting and northing. The correlation between them in reintroduced later by the covariance of the input uncertainty.

To use GPR, two sets of time points $\mathbf{t}_1$ and $\mathbf{t}_2$ have to be specified and corresponding values of the mean ($m(\mathbf{t}_1)$ and $m(\mathbf{t}_2)$) and covariance ($k(\mathbf{t}_1, \mathbf{t}_1)$, $k(\mathbf{t}_2, \mathbf{t}_2)$ and $k(\mathbf{t}_1, \mathbf{t}_2)$) computed. Let $\mathbf{t}_1$ be the times from where training points are from and $\mathbf{t}_2$ the set of times, where the regression function will be evaluated. The coordinates of the training points are $\mathbf{y}_E$ and $\mathbf{y}_N$ and the corresponding coordinates for the times, where the regression is evaluated at are $\mathbf{y^*}_E$ and $\mathbf{y^*}_N$. These values come from a multivariate Gaussian distribution:

$$\begin{pmatrix} \mathbf{y}_E \\ \mathbf{y}_N \\ \mathbf{y^*}_E \\ \mathbf{y^*}_N \end{pmatrix} \sim \mathcal{N}\left(\begin{pmatrix} m_E(\mathbf{t}_1) \\ m_N(\mathbf{t}_1) \\ m_E(\mathbf{t}_2) \\ m_N(\mathbf{t}_2) \end{pmatrix}, \begin{pmatrix} k_E(\mathbf{t}_1, \mathbf{t}_1) & \mathbf{0} & k_E(\mathbf{t}_1, \mathbf{t}_2) & \mathbf{0} \\ \mathbf{0} & k_N(\mathbf{t}_1, \mathbf{t}_1) & \mathbf{0} & k_N(\mathbf{t}_1, \mathbf{t}_2) \\ k_E(\mathbf{t}_2, \mathbf{t}_1) & \mathbf{0} & k_E(\mathbf{t}_2, \mathbf{t}_2) & \mathbf{0} \\ \mathbf{0} & k_N(\mathbf{t}_2, \mathbf{t}_1) & \mathbf{0} & k_N(\mathbf{t}_2, \mathbf{t}_2) \end{pmatrix} \right) \tag{27}$$

The distribution defined by equation (27) is actually two independent multivariate distributions: one for easting and other for northing. If the covariance in easting and northing is not interesting, then these two can be split up, but in this work dependence between them is restored when the observations are included. For convenience the notation can be simplified by merging the easting and northing:

$$\begin{pmatrix} \mathbf{y} \\ \mathbf{y}^* \end{pmatrix} \sim \mathcal{N}\left(\begin{pmatrix} \mathbf{m}_1 \\ \mathbf{m}_2 \end{pmatrix}, \begin{pmatrix} \mathbf{K}_{11}, \mathbf{K}_{12} \\ \mathbf{K}_{21}, \mathbf{K}_{22} \end{pmatrix} \right).$$

When the values of $\mathbf{y}$ are known, then these values can be used to update the distribution for $\mathbf{y}^*$:

$$\mathbf{y}^* \sim \mathcal{N}(\mathbf{m}^*, \mathbf{K}^*), \tag{28}$$
$$\mathbf{m}^* = \mathbf{m}_2 + \mathbf{K}_{21}\mathbf{K}_{11}^{-1}(\mathbf{y} - \mathbf{m}_1), \tag{29}$$
$$\mathbf{K}^* = \mathbf{K}_{22} - \mathbf{K}_{21}\mathbf{K}_{11}^{-1}\mathbf{K}_{12}. \tag{30}$$

These equations would be correct, if the values of $\mathbf{y}$ are perfectly known. In the context of this work the values of $\mathbf{y}$ are the predicted locations from the switching Kalman filter and they have covariance $\mathbf{\Sigma}$ associated with them. The matrix $\mathbf{\Sigma}$ contains the error variances for both coordinates for each time step and also the covariance between the easting and northing for each time step. Adding this to the covariance matrix $\mathbf{K}_{11}$ will result in new equations describing the distribution of $\mathbf{y}^*$:

$$\mathbf{m}^* = \mathbf{m}_2 + \mathbf{K}_{21}(\mathbf{K}_{11} + \mathbf{\Sigma})^{-1}(\mathbf{y} - \mathbf{m}_1), \tag{31}$$
$$\mathbf{K}^* = \mathbf{K}_{22} - \mathbf{K}_{21}(\mathbf{K}_{11} + \mathbf{\Sigma})^{-1}\mathbf{K}_{12}. \tag{32}$$

The final thing that needs to be done, before GPR can be applied, is defining the covariance function $k(t, t')$, that is also called kernel. Although there are infinitely many kernels, not every function is a valid covariance function. Proving that a function is a valid kernel, is not a trivial task and is out of the scope of this work.

In this work, two known kernels are used: the squared exponential (SE) covariance function and a periodic kernel. Both of these are isotropic kernels, meaning that they depend only on the absolute value of time difference $|t_1 - t_2|$.

The square exponential kernel is defined as

$$k_{SE}(r) = a \exp(-\frac{r^2}{2l^2}) \tag{33}$$

and has two parameters: the scale parameter a and the length scale l. Because of its simplicity, it is the most commonly used kernel. The periodic kernel

$$k_P(r) = a_p \exp(-\frac{2\sin^2(\frac{r}{T}\pi)}{l_p^2}) \tag{34}$$

has a similar form, but one extra parameter - period T. An important detail, that is often omitted is that the length scales l and l_p have different units. In the square exponential kernel, l has the same units that r has, in this case, units of time. In the periodic kernel l_p is a unitless quantity. There is no single correct method to relating these two quantities. One option is to leave them to be independent and optimize them separately. Another approach would be to make the main peak of the periodic kernel have the same shape as the SE kernel. To achieve this the values of kernels with the same scale parameters ($a = a_p$) can be equated at value $r = l$ and the resulting relationship between l_p and l is

$$l_p = 2\sin(\frac{l}{T}\pi). \tag{35}$$

Although these kernels will equal only at points $r = 0$ and $r = l$, they are visually indistinguishable at the main peak, as seen in Figure 5. The difference between them is less than 1 %. This property holds only when l is much smaller than T.

Existing kernels can be combined to form new ones: the sum of any valid covariance function is also a valid covariance function. The sum of periodic and squared exponential kernels corresponds to a periodic regression model that puts more emphasis on local evidence (time-wise) and less on evidence from other periods. The weight of these is controlled by the scale parameters a and a_p. This combined kernel will still be referred to as a periodic kernel.

3.1.2 Correlated Measurement Errors

In section 2.2.3 correlated measurement errors were covered in the context of Kalman filtering. Even though the correlation aware model was applied to the data, the resulting predictions for the location will still have the same correlation in them. This has to be accounted for when using the Gaussian process regression. Similarly to the SKF, here also it will be assumed, that the correlation mainly affects measurements of stationary devices. Unlike the SKF there is no separate stay model, but the stop episodes have been identified by the SKF. Among these stop episodes, clusters of

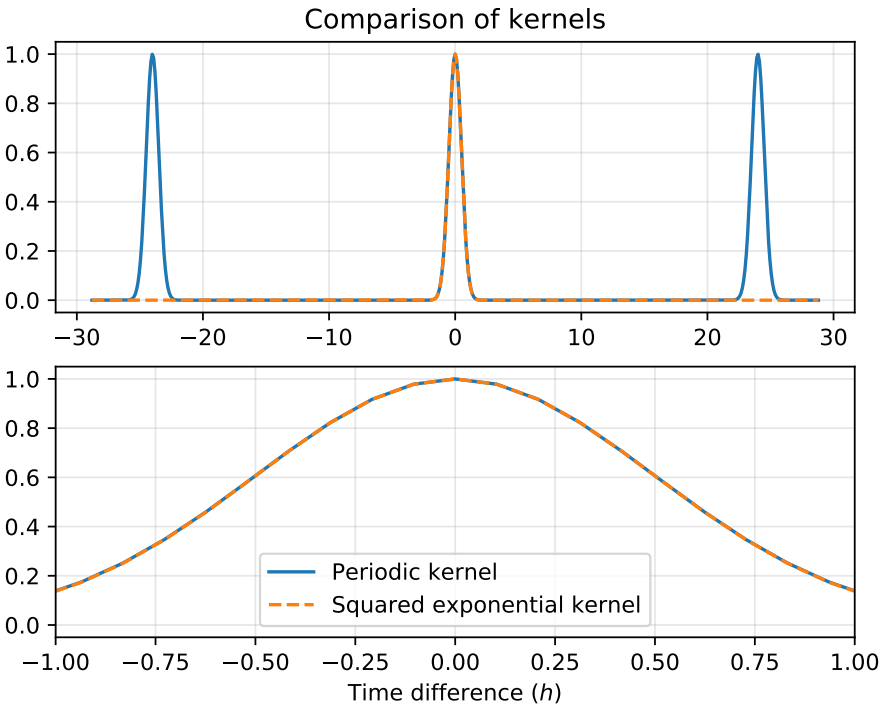

Figure 5. Periodic and squared exponential kernel with unit scales and length scales related by the equation (35). Period is 24 h and the length scale $l = 30$ min.

very similar predictions can be detected. It is assumed that in each of these clusters all the measurement errors are correlated.

This correlation can be included into the model by changing the values of $\mathbf{\Sigma}$ in equations (31) and (32). Currently, there are the estimated uncertainties for each time step for both easting and northing. As an improvement, the correlations among the detected clusters will be also included. Setting the correlation value close to 1 makes the inference algorithm numerically unstable. Correlation values below 0.5 resulted in a covariance matrix that was numerically stable. Using such low correlation values is an improvement over using no correlation but not enough to avoid the uncertainties in the predictions getting unreasonably low. The reasons for this numerical instability and methods to avoid it must be investigated in future works.

3.2 Mixture of Gaussian Process Regressors

Although a large part of human mobility follows a periodic pattern, there are still mobility episodes that are completely out of the pattern. Using Gaussian process regression with the periodic kernel on these aperiodic episodes can be detrimental to the whole regression accuracy. A generated example of this case is shown in Figure 6: the regression line in the aperiodic part gravitates towards the measurements of the periodic parts. A way to tackle this problem would be using two GPR models: one that has only aperiodic kernel and other has a periodic kernel. These models would

be used only for regions that are appropriate for them. A known tool for this is Gaussian process regression mixture model proposed by Volker Tresp [31]. Similarly to most other mixture models it uses an iterative expectation maximisation algorithm for finding which model corresponds to what part of the data. As comparison a regression line of mixture model is added to Figure 6. Its fit is significantly better than a plain periodic model, especially for the aperiodic part. This mixture model is not fitted with the algorithm proposed by Tresp, but by technique proposed in section 3.3.

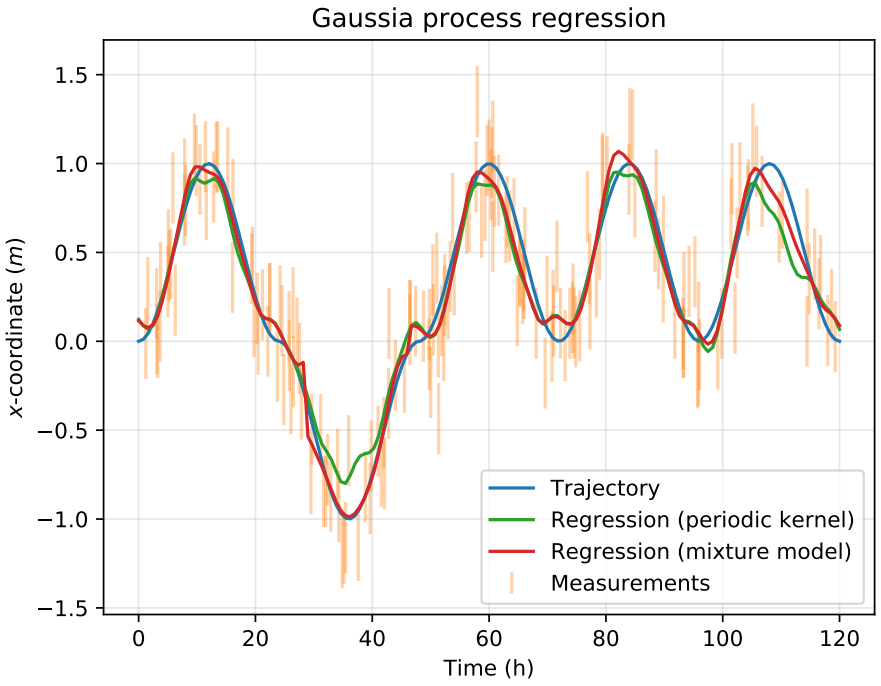

Figure 6. One dimensional trajectory of an object is recorded over 5 days, on four of them its coordinate follows a periodic pattern, but in one it deviates from this pattern. Given noisy measurements, two regression models are fitted: one GPR with a periodic kernel and other is GPRMM of periodic and aperiodic models.

3.3 Hidden Markov Model Based Gaussian Process Regression Mixture Model

The mixture model algorithm introduced by Tresp requires three sets of Gaussian process regressors to be fitted iteratively [31]. This approach is far too slow when hundreds of thousands of trajectories have to be analysed. In this section, an alternative method for fitting GPRMMs is proposed. There are slight differences when applying this method for periodic and aperiodic kernels. These differences are discussed in the following subsections.

A key point for solving this task is estimating how well is the value of a training point predicted by all other training points. This process is called leave one out

cross-validations (LOO-CV). For many machine learning algorithms, this requires refitting the model n times, where n is the number of training points. Fortunately, for Gaussian process regression, this can be done by fitting the model on the data only once [29, Chapter 5.4.2]. In the context of this work, when there are n training points then the vector $\mathbf{y}$ has length $2n$ because each training point has both easting and northing. The difference between LOO estimate for i-th point and the measured value at that point is also a Gaussian distribution with mean vector

$$\left[[(\mathbf{K}_{11}+\mathbf{\Sigma})^{-1}]_{(i,i+n),(i,i+n)}\right]^{-1}[(\mathbf{K}_{11}+\mathbf{\Sigma})^{-1}(\mathbf{y}-\mathbf{m}_1)]_{i,i+n} \tag{36}$$

and covariance matrix

$$\left[[(\mathbf{K}_{11}+\mathbf{\Sigma})^{-1}]_{(i,i+n),(i,i+n)}\right]^{-1}. \tag{37}$$

The matrix $(\mathbf{K}_{11}+\mathbf{\Sigma})^{-1}$ and vector $(\mathbf{K}_{11}+\mathbf{\Sigma})^{-1}(\mathbf{y}-\mathbf{m}_1)$ both contain only the training points and are already computed during fitting the model. Indices $i, i+n$ mean taking a subvector (with length 2) or a submatrix (with shape 2x2) of these. The computational cost of this is negligible compared to finding the inverse of $\mathbf{K}_{11}+\mathbf{\Sigma}$. The value of this Gaussian at the origin is the likelihood that this point follows this model. By comparing these likelihoods for different models the one can be chosen for each point.

3.3.1 Aperiodic Kernels

Gaussian process regression mixture models with aperiodic kernels have been used to model functions whose behaviour differs for different input values - narrower kernels are better at capturing quicker changes and wider kernel behave better are smoother areas. In the article by Tresp observation from step function were modelled by mixture model with kernels that had three different length scales [31]. A reconstruction of this example is depicted in Figure 7. The mixture achieves better results than all of its component models.

The proposed method for fitting a GPRMM with aperiodic kernels is:

1. All of the GPR models are fitted on the entire training data.
2. LOO-CV log-likelihoods are computed for each training point for each GPR model.
3. A hidden Markov model (HMM) is constructed to model this system [32]. The log-likelihoods are the emission scores for each training point and the transition probabilities are estimated using a continuous time Markov chain, like in the section 2.2.1.
4. The Viterbi algorithm is used to find the most probable model for training point [33]. In between consecutive points with the same model, it is assumed that the model does not change. There can be different strategies used to select model between input points where a transition from one model to other happens - here the transition is chosen to always happen in the middle of these points.

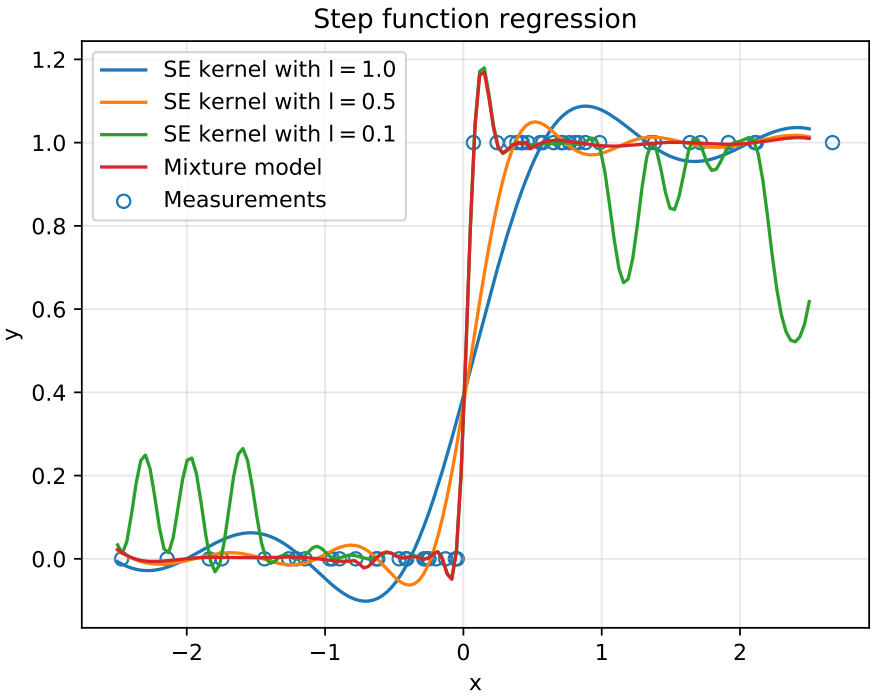

Figure 7. Recreation of a toy example of fitting a step function with three different GPRs and a GPRMM of them. It is unknown which type of kernels were originally used - here square exponential kernels are used. The regression line is overshooting the step, this hints that the kernel might be too smooth for this problem [29, Chapter 4.2.1].

5. Models can be refitted only on the inputs they describe, to speed up the inference process for later uses. It is advisable to include some training points surrounding the models' domain to avoid edge effects of the regression.

This algorithm results in hard clustering - meaning that every input corresponds to only one model, not to a weighted sum of several. Unlike the classical mixture model algorithm, this does not require an iterative fitting of the models, resulting in several times faster runtime. There is no guarantee that this gives optimal GPRMM, but empirically the results are just as good as the original algorithm when using only aperiodic kernels.

If possible, the transition rates of the generator matrix used in the continuous time Markov chain should be estimated using some domain knowledge. Alternatively, there are some iterative methods for estimating them, but this might nullify the performance gain of this algorithm [34].

3.3.2 Kernels With Periodic Components

While it is good to have an algorithm that works with aperiodic kernels, in this work it is important to have a method, that can make an optimal choice between a periodic and aperiodic kernel. When applying previously introduced algorithm in

such case, then most likely the periodic model will be excluded from the final model entirely. When a periodic model is trained on the entire dataset it will give worse predictions because parts of the training data that do not follow the periodic pattern pollute the model. The model tries to average up all the periods. This effect can be seen in Figure 6.

To tackle this problem a special case of the previously proposed algorithm has to be designed. Here the mixture model will contain only two regressors: one with an aperiodic kernel and other with a periodic kernel. The new algorithm consist of the following steps:

1. The aperiodic model is fitted on the entire training data.
2. This model is then sampled at evenly spaced times such that each period of the periodic kernel will have the sampling points at the same times. The number of sampling points is a free parameter of this algorithm.
3. The sampling results are shifted by one period and overlapped with itself. The shifted and unshifted data points are compared pairwise and the likelihood of them overlapping is be computed. The threshold for the overlapping is another free parameter of this algorithm. Both points in a pair whose likelihood is above the threshold are marked as potentially periodic.
4. Points in the training data that are between two potentially periodic points are marked as potentially periodic.
5. The periodic model is fitted on the potentially periodic training points.
6. LOO-CV log-likelihoods are computed for each training point for each GPR model. If a point was not part of the training points for the periodic model, then the training value is simply compared with the predicted value.
7. Similarly to the previously proposed algorithm, a HMM is constructed, then the Viterbi algorithm is applied and finally both models are refitted. This time the refitting is not optional because some training points that were not marked as potentially periodic could still be included in the final training set of the periodic model.

This version of the mixture model fitting algorithm was used to fit the mixture model in Figure 6.

4 Results and Discussion

4.1 Data Description

To assess the effectiveness on the developed tools, a sample trajectory is needed. This trajectory must contain both the mobile network events and also accurate location data, such as GPS track. For this work, a single person was tracked over a period of one month. During that time the subject travelled in an area with a radius of 35 km. There are in total 40 000 mobile events made by its device, with an average gap of 67 s between them. The distribution of time differences between subjects network event is depicted in Figure 8. This subject is a bit more avid mobile user than the average - for the entire population 95 % of events happen with less than 5 min after previous, but for this subject, it is 3.5 min.

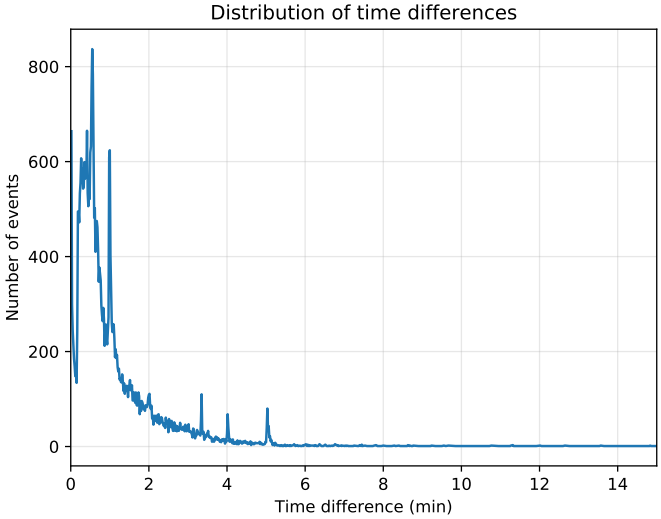

Figure 8. Distribution of time differences between consecutive successful network events for the subjects mobile device. This does not have as prominent tail as the entire population depicted in Figure 1.

This person moves mainly along the shore of a body of water. This makes its trajectory rather anisotropic: 90 % of its location variance is explained by its first principal component. Because of this, on the following visualizations, only its movement in the direction of the first principal component can be shown without any significant loss in information. To preserve the anonymity of the subject and the mobile operator, absolute coordinates are avoided and all the coordinates are deviations from the trajectories' centroid.

4.2 Switching Kalman Filter

The main purpose of using the Switching Kalman filter is detecting if the subject is moving or not. Improved location accuracy is a secondary result of this algorithm.

For this reason, the quality of the model is assessed by its ability to correctly locate the times when the subject was stationary. The probabilities ($\hat{p}_{\text{stop}_t}$) calculated from the SKF do not describe the observation times t, but the transitions from previous observation $t-1$ to this. For each of these transitions from the GPS track, it is found if the device was moving during that time or not. The probability from GPS (p_{stop_t}) is either 0 or 1. These probabilities are combined into a time-averaged log loss:

$$-\frac{\sum_t (p_{\text{stop}_t} \log(\hat{p}_{\text{stop}_t}) + (1 - p_{\text{stop}_t}) \log(1 - \hat{p}_{\text{stop}_t}))\Delta t}{\sum_t \Delta t}. \tag{38}$$

4.2.1 Parameter Optimization

The Switching Kalman filter has two set of parameters: the values of the generator matrix, that determine the values of the model switching matrix $\mathbf{Z}(\Delta t)$ and the value of the process noise covariance matrices $\mathbf{S}$. The values of generator matrix $\mathbf{A}$ are computed using the results of statistical research papers and are shown in equation (14). Each of the movement type model (Stay, Jump, Move) has its own unique process noise covariance matrix. To avoid having fit too many parameters, the same assumptions are made, that were proposed by Batrashev et al [19]:

- The model is isotropic - the uncertainness in the easting and northing will grow in time at the same rate.
- The covariance matrix will have non-zero values only on its main diagonal.
- Stay and Jump models will have the same noise for speed.
- Stay and Move models will have the same noise for location.

These assumptions reduce the number of parameters to four: two location noises and two speed noises.

The trajectory was split into two equal length parts - on the first half the parameter optimization was done and on the second half, the model performance was evaluated. As a result of parameter optimization, the following values resulted in the minimal log-loss:

$$S_{\text{stay}} = \begin{pmatrix} 96\,\frac{\text{m}^2}{\text{s}} & 0 & 0 & 0 \\ 0 & 96\,\frac{\text{m}^2}{\text{s}} & 0 & 0 \\ 0 & 0 & 6.8\,\frac{\text{m}^2}{\text{s}^3} & 0 \\ 0 & 0 & 0 & 6.8\,\frac{\text{m}^2}{\text{s}^3} \end{pmatrix} \tag{39}$$

$$S_{\text{jump}} = \begin{pmatrix} 1500\,\frac{\text{m}^2}{\text{s}} & 0 & 0 & 0 \\ 0 & 1500\,\frac{\text{m}^2}{\text{s}} & 0 & 0 \\ 0 & 0 & 6.8\,\frac{\text{m}^2}{\text{s}^3} & 0 \\ 0 & 0 & 0 & 6.8\,\frac{\text{m}^2}{\text{s}^3} \end{pmatrix} \tag{40}$$

$$S_{\text{move}} = \begin{pmatrix} 96\,\frac{\text{m}^2}{\text{s}} & 0 & 0 & 0 \\ 0 & 96\,\frac{\text{m}^2}{\text{s}} & 0 & 0 \\ 0 & 0 & 22\,\frac{\text{m}^2}{\text{s}^3} & 0 \\ 0 & 0 & 0 & 22\,\frac{\text{m}^2}{\text{s}^3} \end{pmatrix}. \tag{41}$$

4.2.2 Effect of the Improvements to the Algorithm

Three independent improvements were proposed to the original algorithm. To get a quantitative understanding of the effects of these, the loss function was evaluated on the second half of the trajectory with all different combinations of these improvements applied to it. These results are summarised in Table 1. It is reassuring to see that the worst performing model is the one with none of the improvements applied and the one with all of them is the best performing model. It is also important to note that adding an additional improvement to any of the models improves the performance.

Table 1. Log-loss values for different variations of the Switching Kalman filter.

Model switching matrix	Process noise	Correlation awareness	Log-loss
Original	Original	Original	0.2089
Original	Original	Improved	0.1889
Original	Improved	Original	0.2011
Original	Improved	Improved	0.1787
Improved	Original	Original	0.2056
Improved	Original	Improved	0.1855
Improved	Improved	Original	0.2006
Improved	Improved	Improved	0.1749

Majority of the performance gains comes from the third improvement - correlation aware stay model. A common example of a situation, where this improvement helps is depicted in Figure 9. Before and after the depicted time there was a long period of correlated measurements making the prediction of the original algorithm have unreasonably low uncertainty in the location of the device. This made the new observations significantly less impactful and the predicted location just drifted from one location to another without opting for the Move model. The improved variant was justifiably much less certain in its prediction and followed the new data more willingly.

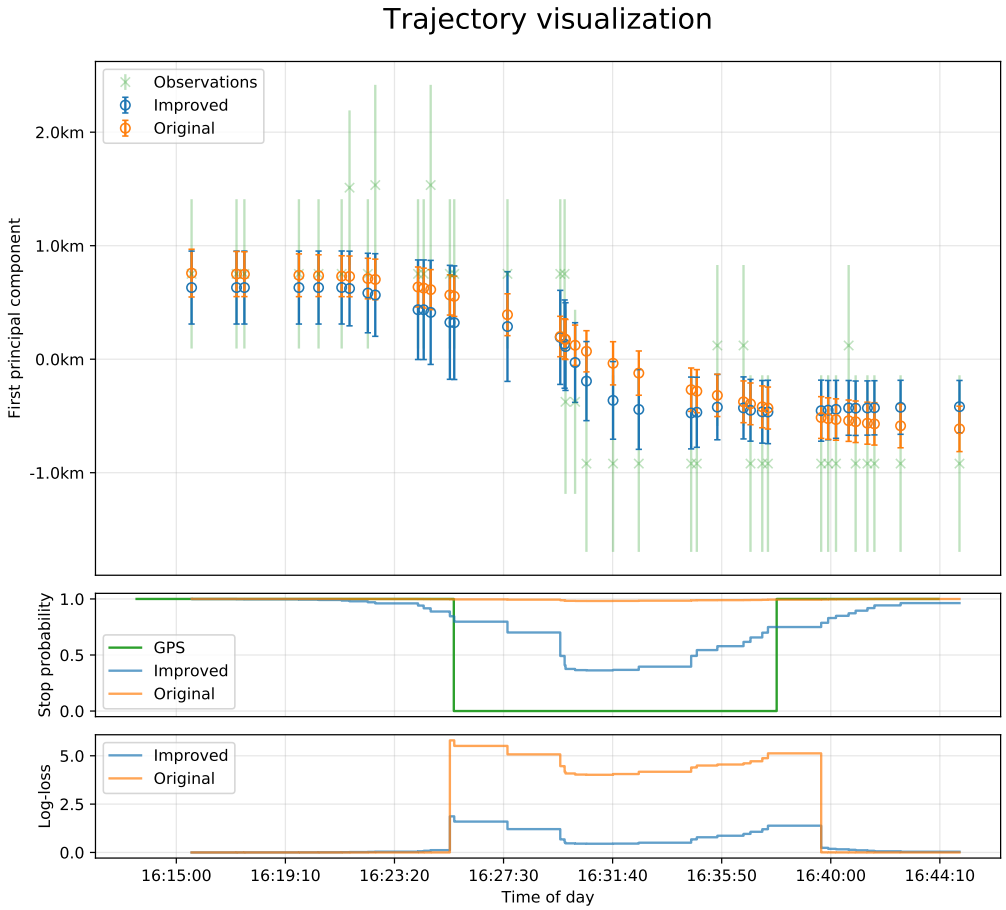

Figure 9. Comparison of the outputs of the original switching Kalman filter and a improvement with correlation aware Stay model.

Making the process noise $\mathbf{Q}$ depend on the time difference Δt was the second-best improvement. For the Stay and Jump models the process noise scales linearly in time (equation (19) and for the Move model the scaling is with a cubic equation (20). When only this improvement is applied, then the major differences between models should occur at time points that have unusually large gaps between them. One of such cases is demonstrated in Figure 10. The original algorithm models this movement as two distinct jumps, the improved model puts more emphasis on the measurements are correctly identified as one continuous movement.

The least significant improvements came from making the model switching matrix $\mathbf{Z}_t$ depend on the time difference Δt. Similarly to the process noise, it has most significant effect in the regions with unusually large Δt - an example of this is depicted in Figure 11. Unlike two previous examples, there is almost no change in the predicted location nor its covariance - only slight change in the model probabilities.

These are the results when the improvements are used one at a time. By using all of them together the score improves even further, but the observations made on the partial result also hold for the complete model. It is important to note, that these changes to the model can in some parts of the data make the prediction slightly

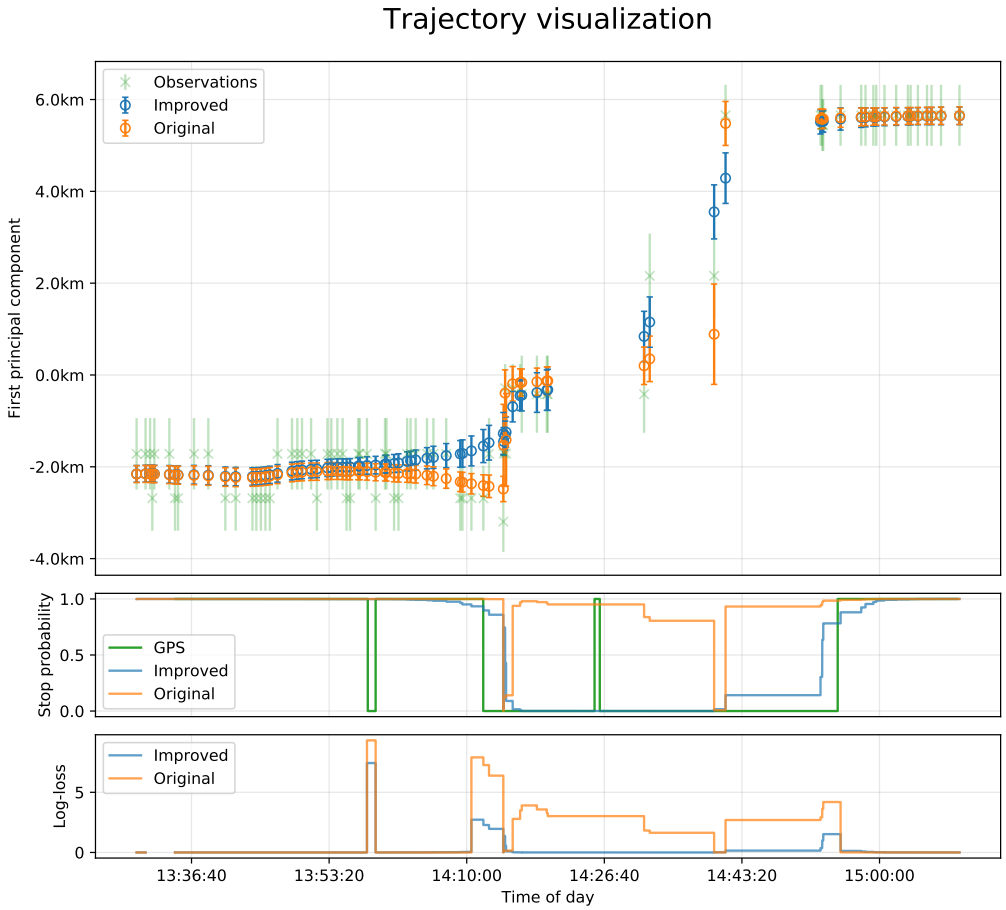

Figure 10. Comparison of the outputs of the original switching Kalman filter and a improvement with sampling rate dependant process noise.

worse, but unlike the improvements, these do not seem to be systematic effects, but rather random fluctuations.

4.3 Gaussian Process Regression Mixture Model

For Gaussian process regression, the main result is the location prediction. To assess the quality of this model, the predictions are compared with the GPS location. Besides the location accuracy, it is also important to have a good estimate of the uncertainty of the prediction. The predictions for any time will be a Gaussian distribution - the additive inverses of logarithms of these distribution values at the GPS points locations will be evaluated and used as the log-loss.

4.3.1 Parameter Optimization

The Gaussian process mixture model contains two regressors: one has a square exponential kernel and other has a kernel that is a weighted sum of a periodic and a square exponential kernels. All of these kernels will have the same length parameter

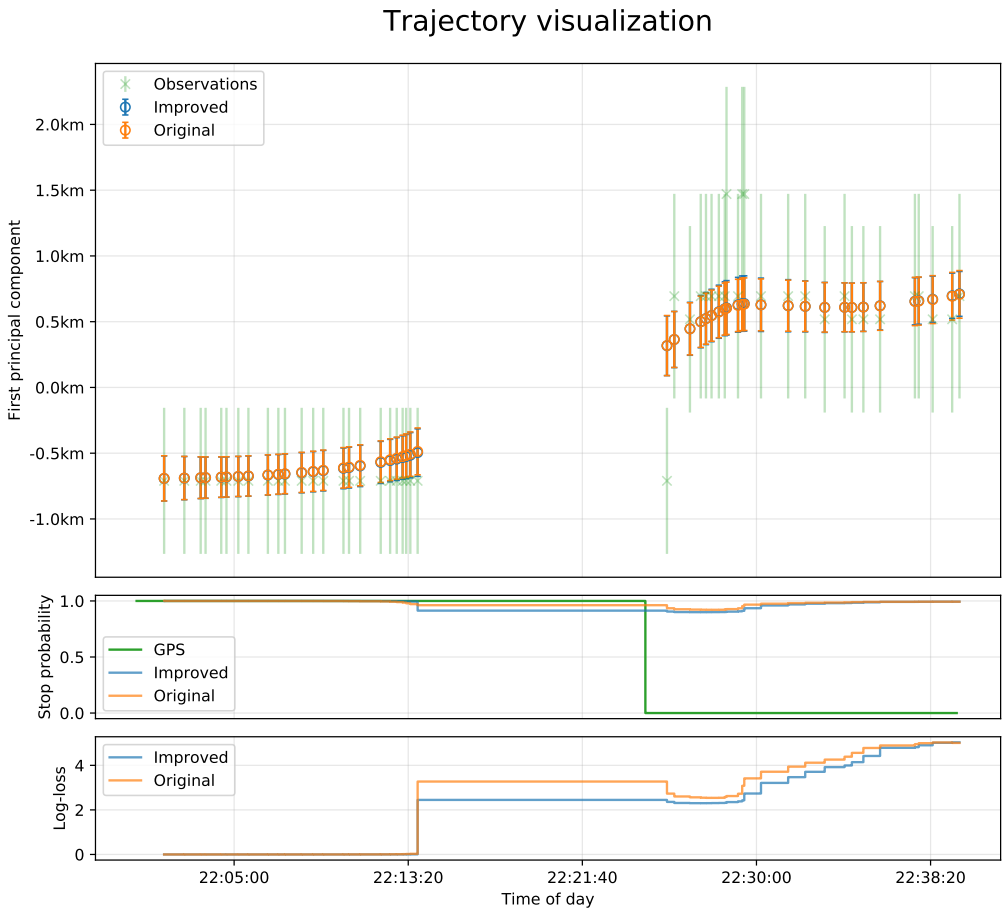

Figure 11. Comparison of the outputs of the original switching Kalman filter and a improvement with sampling rate dependant model switching probabilities.

- for the periodic, the correction in equation (35) has to be applied. This model will have two parameters that have to be optimized: the length scale for kernels and the weight for the periodic kernel. Half of the trajectory is used to find parameters that minimize the average log-loss. The optimal for the length scale is 1.5 h and the optimal weights are such that the central peak is 30 % higher than all the others.

4.3.2 Effects on Location Accuracy

After the optimal parameter values are found, the model can be applied to the rest of the data and the results compared to the GPS track. In the Table 2 the mixture model is compared to the results of a GPR with just a square exponential kernel and the results of just using the Switching Kalman filter. For both of the Gaussian process models, the inputs are the values from the Switching Kalman filter. Because the log-loss values are not intuitive to interpret, the average distance between the predictions and real locations is also included.

Firstly it seems that the regression model does not have any effect on the location accuracy. The 20 m change in the values is in the range of noise and there is no

Table 2. Log-loss values for models predicting the location of a mobile device.

Model	Average log-loss	Average distance
Kalman filter	22.4	420 m
GPR	41.2	440 m
GPRMM	43.7	420 m

systematic improvement. The values of the loss function get significantly worse, when using the GPR model and even more with GPRMM. The main reason for this is correlated inputs, that make the prediction unrealistically confident. While it was partially compensated, it is clearly not enough and more work has to be done with it to use this regression model as a viable method for trajectory analysis.

Besides location predictions, it might be also interesting to take a look at when did the periodic model apply and when the aperiodic one. The device followed the periodic model 80 % of the time. Length of an average periodic pattern was 25 h, aperiodic episodes were much shorter - 2 h on average. The longest periodic sequence was almost 4 days and the longest aperiodic was 12 h. Most of the aperiodic episodes started after the end of a workday: between 5 pm and 8 pm, and ended around 11 pm. While this model did not add any value to the location accuracy, the changes between periodic and aperiodic behaviour might be an interesting future research subject.

5 Conclusion

Mobile operators have access to a sizeable source of data about its clients' mobility. Its vast amounts and potential business value make it appealing to researchers and entrepreneurs alike. Its spatial coarseness and temporal sparseness makes extracting information from this data a compelling task requiring specially crafted algorithmic tools. Detecting when and where did the mobile device stopped is a crucial step that serves as a basis for subsequent data analysis tasks on this data.

In this work, a particular method for stop detection was studied and three shortcomings of it were identified. Potential solutions, that based on the peculiarities of the mobile network, were proposed to these problems. The effectiveness of these solutions was evaluated on a real data. Each of these three changes individually improved the performance and combining them resulted in an even better model.

As a potential way to improve the accuracy of the location predictions, using the prominent periodic patterns of human mobility was investigated. During this, an alternative approach to fitting Gaussian process regression mixture models was developed and tested. Due to the inability to sufficiently include the correlation of inputs into the model, the results did not improve compared to the baseline. Despite this, it was shown that periodicity is a common property of trajectories. Developing the algorithms further so that they can handle correlated inputs is a potential future research topic.

6 Acknowledgments

I would like to express my gratitude to my supervisor Toivo Vajakas, who introduced me to the exciting field of mobile positioning and whose criticisms of the state of the art works laid the foundation of my thesis.

This research has been supported by European Regional Development Fund under the grant no EU48684.

References

[1] Y. Zhang, N. Liu, Z. Pan, T. Deng, and X. You, "A fault detection model for mobile communication systems based on linear prediction," in *2014 IEEE/CIC International Conference on Communications in China (ICCC)*, pp. 703–708, 2014.

[2] S. Rezaei, H. Radmanesh, P. Alavizadeh, H. Nikoofar, and F. Lahouti, "Automatic fault detection and diagnosis in cellular networks using operations support systems data," in *NOMS 2016 - 2016 IEEE/IFIP Network Operations and Management Symposium*, pp. 468–473, 2016.

[3] M. C. González, C. A. Hidalgo, and A. L. Barabási, "Understanding individual human mobility patterns," *Nature*, vol. 453, no. 7196, pp. 779–782, 2008.

[4] O. Järv, R. Ahas, and F. Witlox, "Understanding monthly variability in human activity spaces: A twelve-month study using mobile phone call detail records," *Transportation Research Part C: Emerging Technologies*, vol. 38, pp. 122–135, 2014.

[5] O. Järv, R. Ahas, E. Saluveer, B. Derudder, and F. Witlox, "Mobile Phones in a Traffic Flow: A Geographical Perspective to Evening Rush Hour Traffic Analysis Using Call Detail Records," *PLOS ONE*, no. 11, pp. 1–12, 2012.

[6] J. Steenbruggen, "Data from mobile phone operators: A tool for smarter cities?," *Telecommunications Policy*, vol. 39, no. 3-4, pp. 335–346, 2015.

[7] V. Kolar, S. Ranu, A. P. Subramainan, Y. Shrinivasan, A. Telang, R. Kokku, and S. Raghavan, "People in motion: Spatio-temporal analytics on Call Detail Records," in *2014 Sixth International Conference on Communication Systems and Networks (COMSNETS)*, pp. 1–4, 2014.

[8] T. Vajakas, J. Vajakas, and R. Lillemets, "Trajectory reconstruction from mobile positioning data using cell-to-cell travel time information," *International Journal of Geographical Information Science*, vol. 29, no. 11, pp. 1941–1954, 2015.

[9] I. Siomina and D. Yuan, "Load balancing in heterogeneous LTE: Range optimization via cell offset and load-coupling characterization," in *Communications (ICC), 2012 IEEE International Conference on*, pp. 1357–1361, 2012.

[10] D. M. Rose and T. Kürner, "Outdoor-to-indoor propagation—Accurate measuring and modelling of indoor environments at 900 and 1800 MHz," in *Antennas and Propagation (EUCAP), 2012 6th European Conference on*, pp. 1440–1444, 2012.

[11] K. Yang, I. Gondal, B. Qiu, and L. S. Dooley, "Combined SINR based vertical handoff algorithm for next generation heterogeneous wireless networks," in *Global Telecommunications Conference, 2007. GLOBECOM'07. IEEE*, pp. 4483–4487, 2007.

[12] M. Amirijoo, P. Frenger, F. Gunnarsson, H. Kallin, J. Moe, and K. Zetterberg, "Neighbor cell relation list and physical cell identity self-organization in LTE," in *Communications Workshops, 2008. ICC Workshops' 08. IEEE International Conference on*, pp. 37–41, 2008.

[13] P. Fiadino, D. Valerio, F. Ricciato, and K. Hummel, "Steps towards the extraction of vehicular mobility patterns from 3G signaling data," *Traffic Monitoring and Analysis*, pp. 66–80, 2012.

[14] J. Schlaich, T. Otterstätter, M. Friedrich, and Others, "Generating trajectories from mobile phone data," in *Proceedings of the 89th annual meeting compendium of papers, transportation research board of the national academies*, 2010.

[15] F. Calabrese, M. Diao, G. Di Lorenzo, J. Ferreira, and C. Ratti, "Understanding individual mobility patterns from urban sensing data: A mobile phone trace example," *Transportation research part C: emerging technologies*, vol. 26, pp. 301–313, 2013.

[16] Y. Ye, Y. Zheng, Y. Chen, J. Feng, and X. Xie, "Mining individual life pattern based on location history," in *Mobile Data Management: Systems, Services and Middleware, 2009. MDM'09. Tenth International Conference on*, pp. 1–10, IEEE, 2009.

[17] J. Krumm, "Real time destination prediction based on efficient routes," tech. rep., SAE Technical Paper, 2006.

[18] L. Xiang, M. Gao, and T. Wu, "Extracting stops from noisy trajectories: A sequence oriented clustering approach," *ISPRS International Journal of Geo-Information*, vol. 5, no. 3, p. 29, 2016.

[19] O. Batrashev, A. Hadachi, A. Lind, and E. Vainikko, "Mobility Episode Detection from CDR ' s Data using Switching Kalman Filter," in *4th International Workshop on Mobile Geographic Information Systems*, pp. 63–69, 2015.

[20] R. E. Kalman, "A New Approach to Linear Filtering and Prediction Problems," *Transactions of the ASME - Journal of basic Engineering*, vol. 82, pp. 35–45, 1960.

[21] H. E. Rauch, C. T. Striebel, and F. Tung, "Maximum likelihood estimates of linear dynamic systems," *AIAA Journal*, vol. 3, no. 8, pp. 1445–1450, 1965.

[22] K. P. Murphy, "Switching Kalman Filters," tech. rep., 1998.

[23] J. R. Norris, *Markov Chains*. Cambridge Series in Statistical and Probabilistic Mathematics, Cambridge University Press, 1998.

[24] K. Manaugh, L. F. Miranda-Moreno, and A. M. El-Geneidy, "The effect of neighbourhood characteristics, accessibility, home-work location, and demographics on commuting distances," *Transportation*, vol. 37, no. 4, pp. 627–646, 2010.

[25] R. W. McQuaid and T. Chen, "Commuting times - The role of gender, children and part-time work," *Research in Transportation Economics*, vol. 34, no. 1, pp. 66–73, 2012.

[26] P. Axelsson and F. Gustafsson, "Discretizing stochastic dynamical systems using Lyapunov equations," *ArXiv e-prints*, pp. 1–17, 2014.

[27] M. Petovello, K. O'Keefe, G. Lachapelle, and E. Cannon, "Consideration of time-correlated errors in a Kalman filter applicable to GNSS," *Journal of Geodesy*, vol. 83, pp. 51–56, 2009.

[28] K. Wang, Y. Li, and C. Rizos, "A new practical approach to Kalman filtering with time-correlated measurement errors," *IEEE Transactions on Aerospace and Electronic Systems*, 2009.

[29] C. E. Rasmussen and C. K. I. Williams, *Gaussian processes for machine learning.*, vol. 14. MIT Press, 2004.

[30] H. Hotelling, "Analysis of a complex of statistical variables into principal components.," *Journal of Educational Psychology*, vol. 24, no. 6, pp. 417–441, 1933.

[31] V. Tresp, "Mixtures of Gaussian processes," *Advances in neural information processing systems*, pp. 654–660, 2001.

[32] L. E. Baum and T. Petrie, "Statistical Inference for Probabilistic Functions of Finite State Markov Chains," *Ann. Math. Statist.*, vol. 37, no. 6, pp. 1554–1563, 1966.

[33] A. Viterbi, "Error bounds for convolutional codes and an asymptotically optimum decoding algorithm," *IEEE Transactions on Information Theory*, vol. 13, pp. 260–269, apr 1967.

[34] W. Wei, B. Wang, and D. Towsley, "Continuous-time hidden Markov models for network performance evaluation," vol. 49, no. August 2002, pp. 129–146, 2002.

Appendix

I. Acronyms

CGI	Cell Global Identity
GPR	Gaussian Process Regression
GPS	Global Positioning System
HMM	Hidden Markov Model
KF	Kalman Filter
MC	Markov chain
MLE	Maximum Likelihood Estimation
MM	Mixture Model
RTS	Rauch, Tung and Striebel (smoothing algorithm)
SE	Square Exponential (kernel)
SKF	Switching Kalman Filter

II. Licence

Non-exclusive licence to reproduce thesis and make thesis public

I, **Tanel Kiis**,

1. herewith grant the University of Tartu a free permit (non-exclusive licence) to:

   1.1 reproduce, for the purpose of preservation and making available to the public, including for addition to the DSpace digital archives until expiry of the term of validity of the copyright, and

   1.2 make available to the public via the web environment of the University of Tartu, including via the DSpace digital archives until expiry of the term of validity of the copyright,

   of my thesis

   Stop Detection and Location Accuracy Improvement in Mobile Positioning

   supervised by Toivo Vajakas

2. I am aware of the fact that the author retains these rights.

3. I certify that granting the non-exclusive licence does not infringe the intellectual property rights or rights arising from the Personal Data Protection Act.

Tartu, 21.05.2018