Spatio-Temporal Data Mining with Event Logs from High Volume Logistics Information

RODRIGO MIGUEL TAVARES GONÇALVES*
Instituto Superior Técnico - Universidade de Lisboa, Lisboa, Portugal
rodrigo.goncalves@tecnico.ulisboa.pt
September, 2015

Abstract
In logistics, software aids for transportation planning and scheduling are often based in approximations and abstractions that do not take into account real-world data. The aim of this work is to provide an analysis and methodology, based on real-world data, on how to obtain probability density functions for prediction of activity duration. Given a large spatio-temporal database of events, where each event consists of the fields event ID, time, location, and event type, the aim is to extract valuable information about activities duration. The process is not straightforward since the log is human-influenced creating uncertainty related with the time at which the events are logged. In order to overcome this, a novel framework is proposed: it uses the spatio-temporal trajectories to identify regions-of-interest based on speed, and builds an ROI activity time-line using the activities extracted from event logs. Keywords: Spatio-Temporal, Event Logs, Logistics, Event Mining, Trajectories.

I. I N T R O D U C T I O N

Event logs provide a chronological record of a sequence of activities that is essential to gather information about complex systems [16]. The problem is that, in many cases, events are introduced by humans, the system users, leading to the existence of uncertainty in the data [7]. When system logs are human dependent, it is not assured that event records happened in coherence with reality. For example, if an activity is characterized by a start and an end event, its duration can be calculated as the time difference between such events. However, if the events are recorded before, or after, such occurrences the previous statement no longer holds true. In order to overcome such problem, a new algorithm that combines spatio-temporal and event logs databases is proposed.

*This work was performed as part of the DAIPEX project grand funded by Dinalog
tremely large sequences, such algorithms were not capable of producing useful results.

II. Spatio-Temporal Event Log Mining

The spatio temporal event log mining (STEL) algorithm, is designed to mine spatio-temporal data bases in conjunction with event logs. STEL is meant to estimate the duration of process activities that are logged on human based event logs. As stated previously, the human behaviour is subject to mistakes. Such human errors create an uncertainty in the event log since events are not assured to be logged in coherence with reality [7]. In other words, users are able to log events before, or after, the corresponding occurrences happening making the duration of logged activities to differ from reality. By using spatio-temporal data its possible to define a time portion of the trajectory where the activities took place, based on the vehicle speed. If the activities that took place in such time-portion are known, its possible to create an activity time-line and through that estimating the duration of the activities. The algorithm is composed by the three main steps that are going to be outlined in the following sections.

I. Regions of Interest

In this work, regions of interest are identified based on the average speed of moving objects. From the user point of view, the concept of trajectory is based in the evolving position (perceived as a point) of an object travelling, in some space, during a given time interval. Thus, a trajectory is by definition a spatio-temporal concept. A GPS trajectory can be formally defined as in [5], [18], [14]:

Definition 1. A trajectory is a finite sequence of space-time points \( \langle p_0, p_1, \ldots, p_N \rangle \), where \( p_i = (\phi_i, \lambda_i, t_i) \) and \( N \) is the total number of data-points in a trajectory. The \( \phi_i, \lambda_i \in \mathbb{R}^2 \) are spatial coordinates, and the \( t_i \in \mathbb{R}^+ \), are timestamps, with \( t_i < t_i + 1 \) for \( i = 0, 1, \ldots, N \).

Each \( (\phi_i, \lambda_i) \) pair represents the position recorded of a moving object at time \( t_i \). A trajectory is then formed by a sequence of segments called trajectory links.

Definition 2. A trajectory link \( l_i \) is a straight line between two consecutive points \( p_i = (\phi_i, \lambda_i, t_i) \) and \( p_{i+1} = (\phi_{i+1}, \lambda_{i+1}, t_{i+1}) \) of the same trajectory, where \( i \in \mathbb{N}_0 \) and \( j = i \).

If an object takes \( \Delta t \) time to travel \( \Delta x \) distance, it maintains an average speed of \( \frac{\Delta x}{\Delta t} \). Using the latitude and longitude of two consecutive data points the distance \( \Delta x \) between them (e.g. length of the link) can be obtained. The average speed of each link can then be calculated using the timestamps of each data point \( p_i \). Conceptually, a region of interest is intended to be a region where moving objects pause or wait in order to complete activities that are difficult or impossible to carry out while in motion. In this work, a region of interest is formulated as follows:

Definition 3. A region \( R \) is a region of interest if at least one trajectory link \( l_i \in R \) of the tracked object has its average speed between \( [0, s_{\text{bound}}] \) and the object remains in \( R \) for at least \( T \) time before leaving \( R \). That is, \( \sum_{i=j}^{n} t_{i+1} - t_i \geq T \) with \( R = [l_j, l_n] \). The parameters \( s_{\text{bound}} \) and \( T \) are user-defined.

In the following figure it is presented the framework used to extract ROIs:

![Figure 1: Framework to extract regions of interest](image)
II. Identify Activities

The event log provides a record of specific events at specific timestamps. Such events can be seen as atomic occurrences, with no time duration. They do not provide an explicit knowledge about activities. Hence, activities must be extracted from the event logs in order to be characterized and studied. Given the large amount of event types present in event logs, it is necessary to do an event log analysis so that the events related to activities and activities them self are identified. From activity related keywords found in the event log, the set of activities can be defined.

\[ A = \{ a_1, a_2, \ldots, a_k \} \]  \hspace{1cm} (1)

Definition 4. An activity \( a_k \), where \( k \) is the index that identify an activity, is a finite sequence data-points \( \langle p_0, p_1, \ldots, p_m \rangle \), where \( p_i = (\phi_i, \lambda_i, t_i, e_i) \), such that \( e_0 = s_k \) and \( e_m = f_k \). The \( \phi_i, \lambda_i \in \mathbb{R}^2 \) are spatial coordinates, the \( t_i \in \mathbb{R}^+ \), are timestamps, with \( t_i < t_{i+1} \) for \( i = 0, 1, \ldots, N \) and \( e_i \) are the recorded events. \( s_k \) and \( f_k \) denote the events that indicate the start and the end of an activity, respectively.

In the same empirical manner, three sets of events are defined: \( S \) - the set of events \( s_k \) that indicate a start of an activity \( a_k \), \( F \) - the set of events \( f_k \) that indicate the end of an activity \( a_k \) and \( C \) - the set of identifiers events \( c_k \) whose presence in the sequence \( a_k \) indicate a special occurrence.

\[ S = \{ s_1, s_2, \ldots, s_k \} \]
\[ F = \{ f_1, f_2, \ldots, f_k \} \]
\[ C = \{ c_1, c_2, \ldots, c_k \} \]  \hspace{1cm} (2)

The lack of correlation between the time at which the events take place and the time that \( s_k \) and \( f_k \) events are logged at leads to a wrongly duration of the logged activity. To overcome this, the STEL algorithm estimates activity durations based on activity time-lines. Time-lines are composed by the start dates \( t_1 \) and end dates \( t_n \) of the regions of interest and the start and end dates of the activities that were performed on those time-spans. Those dates are given by the date at which the \( s_k \) and \( f_k \) events were logged, respectively.

![Figure 2: Activity time-line of a region of interest](image)

III. Characterize Activities

Using the obtained activity time-lines it is possible to estimate, under some assumptions, the time duration for the process related activities. The dependence on humans to log certain events leads to empty times in activity time-lines that should not exist. If a system can keep track of all activity types, the activity time-lines should be fulfilled since there is also something happening. For instance, if one thinks on a person day-to-day life, there is no emptiness in what concerns activities. Either we are working or sleeping or waiting, etc. We are always performing an activity. The only reason for that not to happen in event logs, is the human dependence characteristic of such logs.

![Figure 3: Empty time: the time available on the neighborhood of an activity in the ROI time-line](image)

Under that assumption, the duration of human introduced activities can be estimated by “stretching” the activity blocks based on the empty time available in the neighbourhood of such activities. However, only human logged activities should have their duration estimated. Hence, a subset of human logged activities \( A^* \) is defined. The events related to the activities that do not belong to the subset \( A^* \) are
assumed to have been recorded without any time difference from reality. Figure 3 shows an example of it. The complete framework of the STEL algorithm is shown in figure 1.

![Figure 4: Activity duration estimation framework](image)

III. Study Case

The logistics data used to test the STEL algorithm was collected from a fleet of logistics trucks from DHL Global Forwarding - Schiphol and Jan de Rijk Logistics. Each data-entry is formed by the ID of the truck, the position: given by latitude and longitude coordinates and a time-stamp. The events are recorded with a description and their corresponding activity ID. The database contains trajectories from trucks preforming activities in Rotterdam and Schiphol areas. There is also a specific truck preforming activities across Germany, Netherlands, Belgium and France and its going to be referred as the international truck.

I. Event Log Data

Event logs provide a record of ephemeral occurrences that can be related to several types of circumstances. The performed activities are embedded into the event logs as a sequence of specific events. Hence, it is necessary to categorize events according to the occurrences that they are related to in order to defined the $A$, $A^*$, $S$, $F$ and $C$ sets. Event classes are defined by the event prefixes. In this study case database, the classes defined by “Start of/End of”, “Cancellation of” and “Activity Midnight” prefixes are activity related categories.

$$A = \{ \text{Arrive, Break, Costs, Drive, Garage, Gas, \ldots} \}$$
$$S = \{ \text{Start of "a_k"} \}$$
$$F = \{ \text{End of "a_k", Cancellation of "a_k"} \}$$
$$C = \{ \text{Activity Midnight "a_k"} \}$$

Activities like log in and sign up, despite needing the interaction of the user, they are not subject to the human behaviour. Since those activities are preformed on the logging system, the system is aware of when the user start and ends such process, keeping record of the events at the correct time. Activities such as load and unload, on the other hand, are not tracked by the system. Their log is completely dependent on the user, thus being subject to mistakes. Those are the activities representing the subset $A^* \subseteq A$ who are going to have their duration estimated:

$$A^* = \{ \text{Load, Unload} \}$$

Once the activities the trajectories are identified, they have to be classified according to the type of situation by analysing the events that make part of the activity.

- **Normal situation** - a normal situation is given by a pair of events "Start of activity" and "End of activity";
- **Cancellation situation** - it is characterised by a "Start of activity" followed by a "Cancellation of activity" events;
- **Midnight activity** - is formed by a set of three events: "Start of activity" "End of activity" and, in between those events, "Activity Midnight".

The importance in distinguish the types of activities situations plays a big role in estimating the time durations of activities. If the cancellation situations are not differed from normal situations the algorithm takes it as an occurred activity, making the time-span between the "start of activity" and "cancellation of activity" events occupied. The classifier "Midnight activity" is used to distinguish activities that were preformed during the night. Their duration patterns are different from the ones preformed during the day due to workforce differences.
II. Spatio-Temporal Data

The spatio-temporal data of the trajectory database provide sampled positions of the object being tracked. The distance between two consecutive positions is calculated, using the haversine formula, equation 3 and, with the time difference between acquisitions, the average speed of the object in between such positions can be obtained using the time-differential method, equation 4.

\[
\Delta s_j = 2r \sin \left( \sqrt{\sin^2 \left( \frac{\phi_i - \phi_j}{2} \right) + \cos (\phi_i) \cos (\phi_j) \sin^2 \left( \frac{\lambda_j - \lambda_i}{2} \right)} \right) \tag{3}
\]

\[
s_j = \frac{\Delta s_j}{\Delta t_j}, \quad \text{where} \quad \Delta t_j = t_{j+1} - t_j \tag{4}
\]

However, due to the GPS accuracy and precision, the recorded latitude and longitude can differ slightly from the real one. Such poisoning errors are amplified through differentiation leading to erroneous speed. This becomes worse when a high output rate is used, since the positional error remains the same but the time interval is decreased. In addition, when data-points are recorded simultaneously \( \Delta t \) becomes zero making equation 4 unsolvable. To tackle this problem, a speed filter was implemented. The filter sets the average speed of link \( l_j \) equal to the average speed of link \( l_{j-1} \) when the temporal distance between the data points is smaller than \( \delta \), a user-defined amount of time.

\[
s_j = \begin{cases} 
  s_j & \text{if } \Delta t_j \geq \delta, \\
  s_{j-1} & \text{else} 
\end{cases} \tag{5}
\]

As presented in the Definition 3 ROIs are defined in terms of the average speed and thus the definition of \( s_{\text{bound}} \) is crucial since it will be the boundary between a trajectory link to be considered as a candidate ROI or not. If a high value is chosen, the duration of the ROIs will be larger than in reality is since it can incorporate a higher number of trajectory links. On the other hand, small \( s_{\text{bound}} \) values would lead to a falsely high number of ROIs with short durations, owing to the previously tackled GPS inaccuracies that create non-zero speed even when vehicles are stopped.

For instance, if an activity is being performed and there is a speed record greater than \( s_{\text{bound}} \), two regions of interest are detected and the activity will be split onto the time-lines of both regions of interest, see figure 5. The value for the \( s_{\text{bound}} \) bound parameter is a trade-off between the total number of regions of interest of the trajectory and the total time spent by the truck inside such ROIs. It should be chosen as low as possible, so that regions of interest are strictly correlated with vehicle stops, but not too low, so that the same activity is not split onto two regions of interest.

In order not to lose information relative to short truck stops, the minimum duration \( T \) was set to zero. This means that every sequence of trajectory links with stopped classification will represent a region of interest as figure 7 illustrates. In such trajectory, three ROI were find: the first is formed by the trajectory links \( \langle l_1, l_2, l_3 \rangle \), the second by \( \langle l_6, l_7, l_8 \rangle \) and the third by \( \langle l_{10} \rangle \).
activity are stretched, the previous existent gap between the activities disappears making the start data of the following activity constrain to its original position. Hence, single activity and multiple activity situations have to be differentiated.

**Single Activity**

In the case of a single activity, the estimation of the duration is pretty straightforward. If no other activities are present in the region of interest, the activity which duration is to be estimated is assumed to have the same duration as the region of interest, see figure 9a. When the such activity is the first activity of the region of interest, its duration is given by the time span from the beginning of the region of interest until the start of the following activity, see figure 9b. Same thing happens for activities that are placed at the end of a region of interest, the duration is given by the time difference between the end of the previous activity and the end of the region of interest, figure 9c. The estimated duration for activities that are in between other activities is given by the elapsed time between the end of the previous activity and the start of the following activity, see figure 9d.

---

**III. Activity Duration Estimation**

Having the activity time-lines built as in figure 2, the duration estimation is done by stretching the activities blocks. However, two different situations arise at this point: to estimate the duration of a single activity or to estimate the duration of followed activities. Since it is a sequential process, the estimation of the first activity of the group would constrain the estimation of the following activities.

---

**Figure 7:** An example on extracting regions of interest with $T = 0$

---

**Figure 8:** Different types of estimation cases

When both start and end dates of the first ac-
Multiple Activities

When facing a multiple activity situation, as in figure 8b, a different approach has to be taken. In this case, the whole interval in between two others activities, whose duration is not to be estimated, is used. Figure 10 illustrates such interval.

![Figure 10: Multiple activities situation: defining the interval for estimation](image)

Only load and unload single activities whose duration is smaller or equal to $\varepsilon$ have its duration estimated, otherwise, the original duration of the activities is kept. $\varepsilon$ is a user defined constant that sets the boundary between the activity duration being estimated or not. Activities whose duration is shorter than $\varepsilon$ are shown in green (short activity), otherwise in blue (long activity). Activities that do not belong to $A^*$ are shown in gray.

Several hypothesis can be formulated to estimate multiple activities durations. Depending on the assumptions made, different results can be achieved. Load and unload activities of a group, despite its original duration, can be assumed to have the same duration that can be obtain by simply diving the total time interval by the number of activities on it. The obtain result is the one in figure 11 and the duration is given by:

$$\text{new\_duration} = \frac{\text{interval}}{\text{# of activities}}$$ (6)

![Figure 11: Hypothesis 1](image)

Another possible approach is to assume that the activities whose duration is bigger than $\varepsilon$, long activities, are in coherence with reality. In this way, similar to what is done for the single activity case, it is only estimated the duration for the short activities, by using the equation 7. The duration of the long load and unload activities, in blue, is kept constant and the short activity, in green, is stretched to fulfil the empty space of the interval.

$$\text{short\_activities\_duration} = \frac{\text{interval} - \text{long\_activities\_durations}}{\text{# of short\_activities}}$$ (7)

![Figure 12: Hypothesis 2](image)

In the previous approaches all activities were considered and assumed to had happen. In many cases, for multiple load and/or unload activities situations, it was noticed that it is common the existence of a long activity followed by one or two short activities; or the other way around. Such pattern can be assumed as an indicator of activities that were logged by mistake of the user. The long activity logged represents the actual preformed load/unload activity. To emulate such situations, two hypothesis are proposed: i) the short events are simply dismissed and kept the same duration for the long activities, see figure 13 or ii) the short events are dismissed and the long activities duration is estimated based on the available time interval, see figure 14. In the latter case, the new duration for the long activities is given by:

$$\text{long\_activities\_duration} = \frac{\text{interval}}{\text{# of long\_activities}}$$ (8)

![Figure 13](image)

![Figure 14](image)
At this point each load and unload has an estimated duration. Still, it is not possible to predict a load or unload activity duration for a specific costumer. In order for the extracted knowledge to be used in software applications for transportation planning, information must be categorized into locations so that it is possible to obtain probability density functions of the estimated service times for specific customers. This can be done using customer locations. Each costumer has a fixed location where trucks perform load and unload activities, hence, estimated service times can be clustered according their latitude and longitude.

IV. RESULTS

The STEL algorithm is able to produce results for both fleets of trucks or single trucks situations. In the first case, it is possible to obtain probability density functions for the estimated durations of load and unload activities, as well as total service durations, for a specific customer with a given location. It is also possible to obtain such functions with a higher level of granularity to predict durations in a wider areas. Following is shown an example of a probability density function, obtain for the KLM Cargo warehouse, using the formulated hypothesis 1. For comparison, the original duration of such activities is also shown in figure 17.
from all the performed activities and the corre-
respondent duration, see figure [18] An increased
awareness of the amount of non-productive
time and productive time when the truck is
stopped can be achieved by evaluating the ac-
tivities performed in the regions of interest, see
figure [19]

![Diagram](image1)

**Figure 18:** International Truck activity time-line for all the event log

![Diagram](image2)

**Figure 19:** Productivity of the international truck

V. Conclusions and Future Work

The STEL algorithm enables the possibility of
having a-priori information about the expected
duration of activities that are performed at spe-
cific locations allowing the creation of probabil-
ity density functions that can be used as input
in in planning algorithms, like vehicle rout-
ing problem, capable of dealing with stochastic
time-windows, [2]. The algorithm can be used
to estimate travel times between costumer loca-
tions by simply redefine the speed limit of
regions of interest. Such results can be of inter-
est for application that tackle the problem of
dealing with stochastic travel times.

The obtain results were very consistent, reveal-
ing that the time constrains applied by time-
windows and the other activities (whose du-
tration was not estimated) create a well con-
ditioned problem leading to similar output
results independently of the hypothesis used
to estimate the duration of multiply activity
situations. The created activity time-lines for
regions of interest, and for complete trajecto-
ries, also allow a better comprehension of how
the trucks are working given an awareness for
problems related to the work-flow.

While STEL supports interactive parametriza-
tion of the mining algorithm, choosing the right
parameters might be a challenge. A next goal
of STEL is to implement fuzzy logic for the clas-
sification of the trajectory links. Apart from
the average speed, the classification can also
be based in additional parameters such as the
average link acceleration and link length.

The creation of an activity log enables the pos-
sibility to apply sequential pattern algorithms
on a sequence of activities rather than events,
producing patterns with a higher degree level.

It would also be of great interest to include
more event types in the duration estimation
of the activities, rather than only looking at a
singular class of events (activity related events).
Perhaps the presence of other events in the
event sequence would help to reduce the un-
certainty related to the log times of the events.

References

[1] R. Agrawal and R. Srikant. Mining se-
quential patterns. In Proceedings of the 11th
International Conference on Data Engineer-
ing, pages 3–14, Taipei, Taiwan, March
1995.

[2] M. Desrochers, J. Desrosiers, and
M. Solomon. A new optimization al-
gorithm for the vehicle routing problem
with time windows. *Operations research*,


patterns in trajectories of moving objects.

Querying and mining trajectory databases
using places of interest. In *New trends in


