Location Prediction Algorithm

Shashank Sharma

August 2018

1 Introduction

This algorithm is designed to predict human locations in a real world scenario. The GPS data is taken as input and the processed using the below algorithm.

The Algorithm has several steps:

- Detect stay-points (also detect start or end of the trajectory)
- Group stay-points to form states
- Calculate hourly weights for the states
- Apply Markov chain for the data available

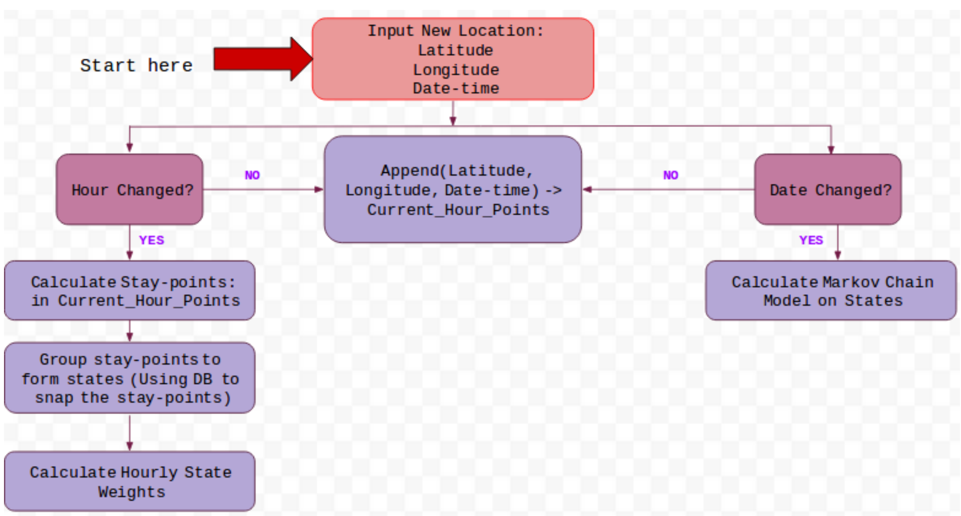

Figure 1: Algorithm Flow-chart

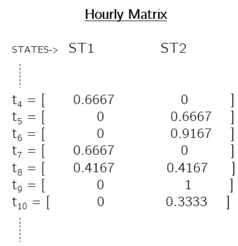

Figure 2: Algorithm Flow-chart

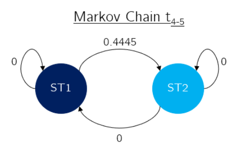

Figure 3: Algorithm Flow-chart

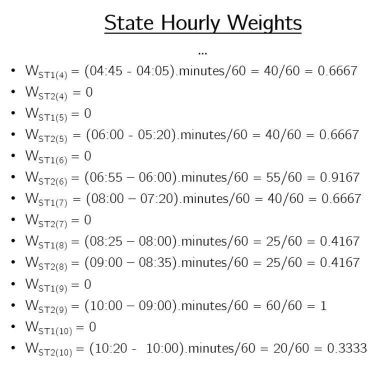

Figure 4: Algorithm Flow-chart

2 Definitions

- **Stay-points:** Stay-points are any points which are stayed by the user in user trajectories or it is the start or the end of the trajectory. For example, if user start at his home, the home itself is a stay-point. Now he move towards work, but he visit a cafe in between for breakfast. The cafe is also a stay-point and then he finishes his trajectory at work, where work is again a stay-point. The places like cafe in this case is identified using distance and time based clustering. For example, a set of points within 200m with total duration of stay greater than 20 minutes can be regarded as a stay-point within the trajectory.

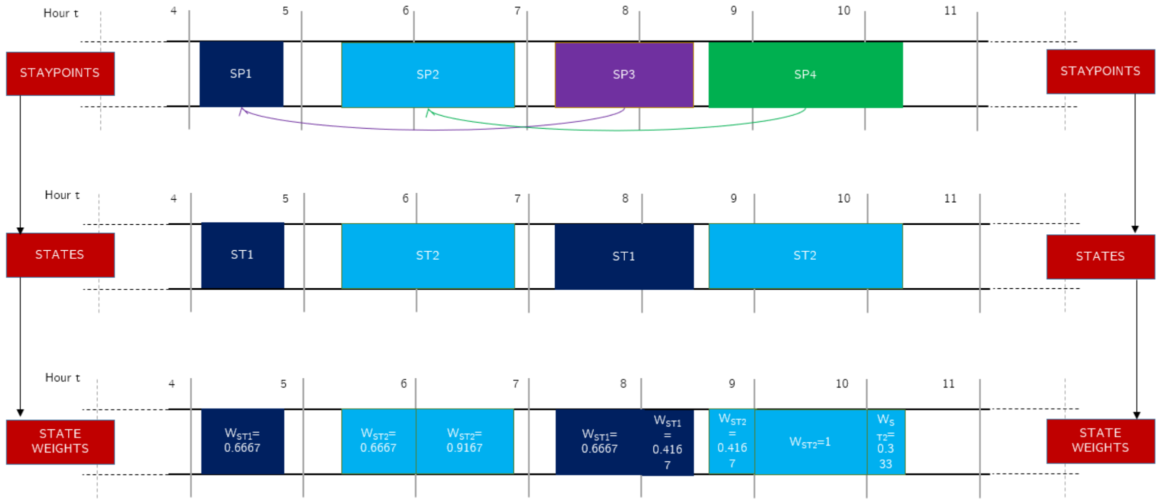

Figure 5: Algorithm Flow-chart

- **State:** A state is formed using a group of stay-points. This is done using a distance threshold for states. All the stay-points within this threshold distance are grouped together as a single state. This is called snapping stay-points to the states. The mean of all location latitudes and longitudes from stay-points within a state are stored per state. Finally Markov Chain model is applied to the states. *Note: A new stay-point is only added to the state if after calculating the mean of the new state, all the existing stay-points still stay within the distance threshold from this mean. This is done to avoid drifting problem while aggregating the stay-points into states.*

3 Algorithms

Algorithm 1 extractStayPoints() : Calculate stay-points

Input: Last hour GPS points *lst_hr_pts[x, y, d]*
Output: Stay-points *sp*

for each *lst_hr_pts[x, y, d]*$_i$ in *lst_hr_pts[x, y, d]* **do**
    if $d_{\mathrm{i}} - d_{\mathrm{i+1}} > th\_tck$ **then**
        $sp \leftarrow lst\_hr\_pts_{\mathrm{i}}, lst\_hr\_pts_{\mathrm{i+1}}$
    else if $distanceBtw(lst\_hr\_pts_{\mathrm{i}}, cluster) <= th\_d$ **then**
        $cluster \leftarrow lst\_hr\_pts_{\mathrm{i}}$
        Update Cluster Mean
    else if $(cluster != empty)$ And $duration(cluster) >= th\_t$ **then**
        $sp \leftarrow cluster$
    end if
end for

Algorithm 2 adjustStartEndStaypoint() : Adjust start-end time of stay-points

Input: Stay-points *sp*
Output: Stay-points with adjusted start and end points *sp*

for each sp_i in *sp* **do**
    $d \leftarrow distanceBtw(sp_{\mathrm{i}}, sp_{\mathrm{i+1}})$
    $t \leftarrow timeDiff(sp_{\mathrm{i}}, sp_{\mathrm{i+1}})$
    $s \leftarrow d/t$
    if $s != 0$ **then**
        $delta\_t \leftarrow min(d,\ th\_d)/s$
    else
        $delta\_t \leftarrow t/2$
    end if
    Update end_time d_e+*delta_t/2* for sp_i
    Update start_time d_s-*delta_t/2* for sp_{i+1}
end for

Algorithm 3 formStates() : Form states from stay-points

Input: Stay-point sp_{n+1} , States st
Output: States st

for each st_j in st **do**
    $is\_sp_{\mathrm{n+1}}\_added \leftarrow False$
    if $distanceBtw(sp_{\mathrm{n+1}}, st_{\mathrm{j}}) <= th\_d$ **then**
        Add $\mathrm{sp_{n+1}}$ to st
        $newMean \leftarrow mean(st)$
        if $distanceBtw(All\ sp's\ of\ st_{\mathrm{j}}, newMean) <= th\_d$ **then**
            Add sp_{n+1} to st_j
            $is\_sp_{\mathrm{n+1}}\_added \leftarrow True$
            Break
        else
            Remove $\mathrm{sp_{n+1}}$ from st
        end if
    end if
end for
if $is\_sp_{\mathrm{n+1}}\_added = False$ **then**
    Add new state in st
end if

Algorithm 4 markovModel() : Create the Markov Model with transition probabilities)

Input: State Weights w
Output: Markov Chain Model mc

for each $h - hour$ from $0to23$ **do**
    if $h! = 23$ **then**
        $trn\_mat_{\mathrm{h+1}} \leftarrow w^{\mathrm{h^{[T]}}} * w^{\mathrm{h+1}}$
        $mc_{\mathrm{h+1}} \leftarrow eachCell(trn\_mat_{\mathrm{h+1}})/rowSum(trn\_mat_{\mathrm{h+1}})$
    else
        $trn\_mat_0 \leftarrow w^{23^{[\mathrm{T}]}} * w^0$
        $mc_0 \leftarrow eachCell(trn\_mat_0)/rowSum(trn\_mat_0)$
    end if
end for