fuzzyMM: Map Matching using a Fuzzy Logic Based Map Matching Algorithm

Nikolai Gorte

March 29, 2014
1 Introduction

Map matching is the process of matching inaccurate GPS trajectories to a digital road network. fuzzyMM is a package that implements a fuzzy logic based map matching algorithm to solve this task. The algorithm consists of three major parts responsible for the identification of the links the vehicle is travelling on:

- Initial MapMatching Process (IMP)
- Subsequent MapMatching Process along a link (SMP-1)
- Subsequent MapMatching Process at a junction (SMP-2)

Each of this processes uses a Fuzzy Inference System (FIS) for the link identification.

A detailed description of the fuzzy logic map matching algorithm and the FIS can be found in Quddus (2006).

2 Usage

2.1 Map Matching

The GPS trajectory must be a SpatialPointsDataFrame. It must include following data to be processed by the algorithm:

- GPS.Speed in km/h
- GPS.Bearing
- GPS.HDOP
- time as “POSIXct” or “POSIXlt”

Additionally a projection must be specified.

```r
> names(track)
[1] "id"          "time"          "GPS.Speed"          
[10] "CO2"          "Consumption"      "Intake.Pressure"     
[16] "GPS.VDOP"      "Calculated.MAF"  "GPS.PDOP"          
[19] "O2.Lambda.Current.ER"  "GPS.Accuracy"   "GPS.Altitude"       
[22] "Long.Term.Fuel.Trim.1"
```

```r
track proj4string(track)
[1] "+proj=longlat +datum=WGS84 +no_defs +ellps=WGS84 +towgs84=0,0,0"
```
As you can see in Figure 1 some of the points of our GPS track are not located on the roads.

Applying the `mm` function uses the fuzzy logic map matching algorithm to reconcile the GPS track with the digital road network.

```r
> matched_track <- mm(track)
```

The result is `SpatialPointsDataFrame` containing the new map matched positions as can be seen in Figure 2.
2.2 Adjusting Membership Functions

It is possible to adjust the membership functions, used in the FIS, to your needs by changing the range of the fuzzy subsets of the input variables. Membership functions are used to fuzzify the input variables of the FIS, which means assigning them to values between 0 and 1. It is recommended to read Quddus (2006) to understand how input variables, fuzzy subsets and membership functions work together.

To see the current range of the fuzzy subsets use

```r
> get_var_bounds()
```

<table>
<thead>
<tr>
<th>variable</th>
<th>left_bounds</th>
<th>right_bounds</th>
<th>ID</th>
</tr>
</thead>
<tbody>
<tr>
<td>speed_high</td>
<td>3</td>
<td>6</td>
<td>1</td>
</tr>
<tr>
<td>speed_low</td>
<td>2</td>
<td>4</td>
<td>2</td>
</tr>
<tr>
<td>speed_zero</td>
<td>0</td>
<td>2</td>
<td>3</td>
</tr>
<tr>
<td>HE_small</td>
<td>20</td>
<td>45</td>
<td>4</td>
</tr>
<tr>
<td>HE_large</td>
<td>25</td>
<td>60</td>
<td>5</td>
</tr>
<tr>
<td>PD_short</td>
<td>10</td>
<td>40</td>
<td>6</td>
</tr>
<tr>
<td>PD_long</td>
<td>20</td>
<td>50</td>
<td>7</td>
</tr>
</tbody>
</table>
HDOP_good 3 5 8
HDOP_bad 4 6 9
alpha_low 85 100 10
alpha_high 90 120 11
beta_low 85 100 12
beta_high 90 120 13
delta_dist_neg -5 5 14
delta_dist_pos -5 10 15
HI_small 10 20 16
HI_large 15 30 17
HI_180 150 200 18
connectivity_direct 0 1 19
connectivity_indirect 0 1 20
dist_err_small 5 15 21
dist_err_large 10 25 22

These bounds represent the x values at which the sigmoidal membership functions reach ≈0 or ≈1 respectively (e.g. “speed is high” ranges from 3 to 6 m/s).

You can change the bounds using `set_var_bounds(name, bounds).

> set_var_bounds("speed_high", c(4, 7))

When everything is set to you needs, update the membership functions and reinitialize the FIS.

> update_mf()

The parameters of the membership functions and all additional information regarding the FIS can be acquired by getting the FIS of IMP, SMP1 or SMP2.

> fis_imp <- get_fis("IMP")
> str(fis_imp)

List of 16
$ num.labels : num [1, 1:4] 3 2 2 2
$ varout.mf : NULL
$ rule : chr [1:6, 1:16] "IF" "IF" "IF" "IF" ...
$ varinp.mf : num [1:5, 1:9] 6 3.06 5.5 NA NA ...
..- attr(*, "dimnames")=List of 2
 .. ..$ : NULL
 .. ..$ : chr [1:9] "high" "low" "zero" "small" ...
$ range.data.ori : num [1:2, 1:5] 0 50 0 360 0 60 0 20 0 100
$ type.model : chr "TSK"
$ type.tnorm : chr "MIN"
$ type.implication.func: chr "MIN"
$ type.mf : chr "SIGMOID"
$ type.defuz : NULL
$ type.snorm : chr "MAX"
$ func.tsk$ : num [1:6, 1] 50 10 50 10 100 10
$ method.type$ : chr "MANUAL"
$ name$ : chr "Sim-1"
$ colnames.var$ : chr [1:5] "v" "HE" "PD" "HDOP" ...
$ class$ :function (x)
- attr(*, "class")= chr "frbs"

> fis_imp$varinp.mf

<table>
<thead>
<tr>
<th></th>
<th>high</th>
<th>low</th>
<th>zero</th>
<th>small</th>
<th>large</th>
<th>short</th>
<th>long</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>6.0</td>
<td>6.0</td>
<td>6.0</td>
<td>6.0</td>
<td>6.0</td>
<td>6.0</td>
<td>6.0</td>
</tr>
<tr>
<td>2</td>
<td>3.1</td>
<td>-4.6</td>
<td>-4.6</td>
<td>-0.4</td>
<td>0.3</td>
<td>-0.3</td>
<td>0.3</td>
</tr>
<tr>
<td>3</td>
<td>5.5</td>
<td>3.0</td>
<td>1.0</td>
<td>32.5</td>
<td>42.5</td>
<td>25.0</td>
<td>35.0</td>
</tr>
<tr>
<td>4</td>
<td>NA</td>
<td>NA</td>
<td>NA</td>
<td>NA</td>
<td>NA</td>
<td>NA</td>
<td>NA</td>
</tr>
<tr>
<td>5</td>
<td>NA</td>
<td>NA</td>
<td>NA</td>
<td>NA</td>
<td>NA</td>
<td>NA</td>
<td>NA</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th></th>
<th>good</th>
<th>bad</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>6.0</td>
<td>6.0</td>
</tr>
<tr>
<td>2</td>
<td>-4.6</td>
<td>4.6</td>
</tr>
<tr>
<td>3</td>
<td>4.0</td>
<td>5.0</td>
</tr>
<tr>
<td>4</td>
<td>NA</td>
<td>NA</td>
</tr>
<tr>
<td>5</td>
<td>NA</td>
<td>NA</td>
</tr>
</tbody>
</table>

It is also possible to plot the membership functions.

> plotMF(fis_imp)
Figure 3: Membership Functions

References