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

Nikolai Gorte

March 12, 2015

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` as defined in `sp` or one of the `Track` classes defined in `trajectories`. 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.

```
> names(track)
```

```
 [1] "id"                     "time"                   "GPS.Speed"
 [4] "O2.Lambda.Current"      "Speed"                  "GPS.HDOP"
 [7] "Short.Term.Fuel.Trim.1" "Rpm"                    "Engine.Load"
[10] "CO2"                    "Consumption"            "Intake.Pressure"
[13] "Throttle.Position"      "GPS.Bearing"            "Intake.Temperature"
[16] "GPS.VDOP"               "Calculated.MAF"         "GPS.PDOP"
[19] "O2.Lambda.Current.ER"   "GPS.Accuracy"           "GPS.Altitude"
[22] "Long.Term.Fuel.Trim.1"
```

```
> proj4string(track)
```

```
[1] "+proj=longlat +datum=WGS84 +no_defs +ellps=WGS84 +towgs84=0,0,0"
```

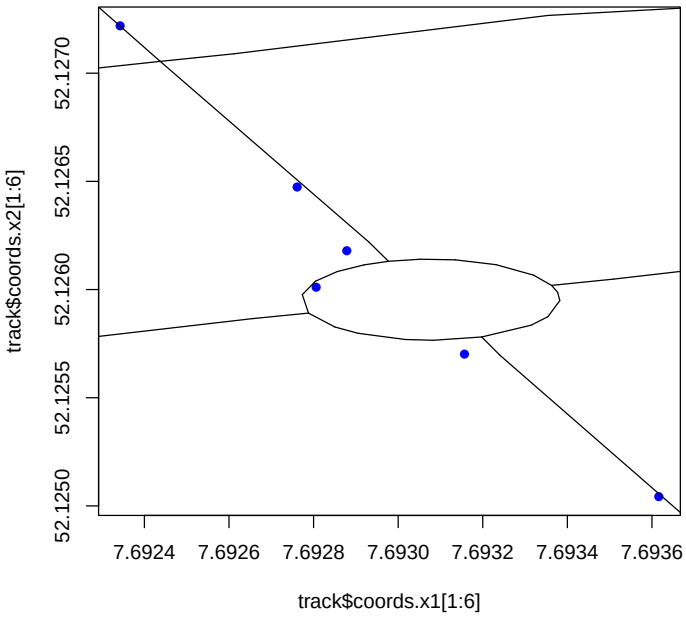

Figure 1: Inaccurate GPS track

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.

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

The result is an object of the same class as the input containing the the new map matched positions as can be seen in Figure 2.

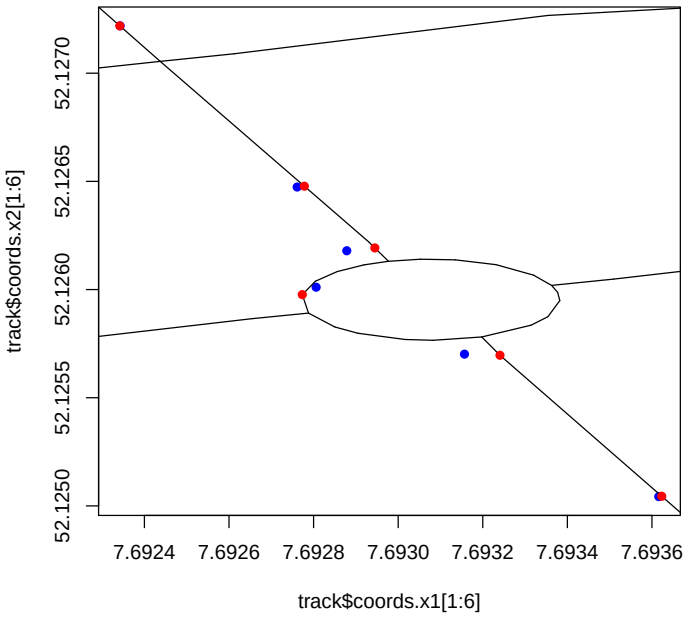

Figure 2: Result after Map Matching

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

```
> get_var_bounds()
```

	left_bounds	right_bounds	ID
speed_high	3	6	1
speed_low	2	4	2
speed_zero	0	2	3
HE_small	20	45	4
HE_large	25	60	5
PD_short	10	40	6
PD_long	20	50	7

```
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, 1:17] "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
```

```
         high      low     zero      small      large      short       long
[1,] 6.000000  6.00000  6.00000  6.0000000  6.0000000  6.0000000  6.0000000
[2,] 3.063413 -4.59512 -4.59512 -0.3676096  0.2625783 -0.3063413  0.3063413
[3,] 5.500000  3.00000  1.00000 32.5000000 42.5000000 25.0000000 35.0000000
[4,]       NA       NA       NA         NA         NA         NA         NA
[5,]       NA       NA       NA         NA         NA         NA         NA
         good     bad
[1,]  6.00000 6.00000
[2,] -4.59512 4.59512
[3,]  4.00000 5.00000
[4,]       NA      NA
[5,]       NA      NA
```

It is also possible to plot the membership functions.

```
> plotMF(fis_imp)
```

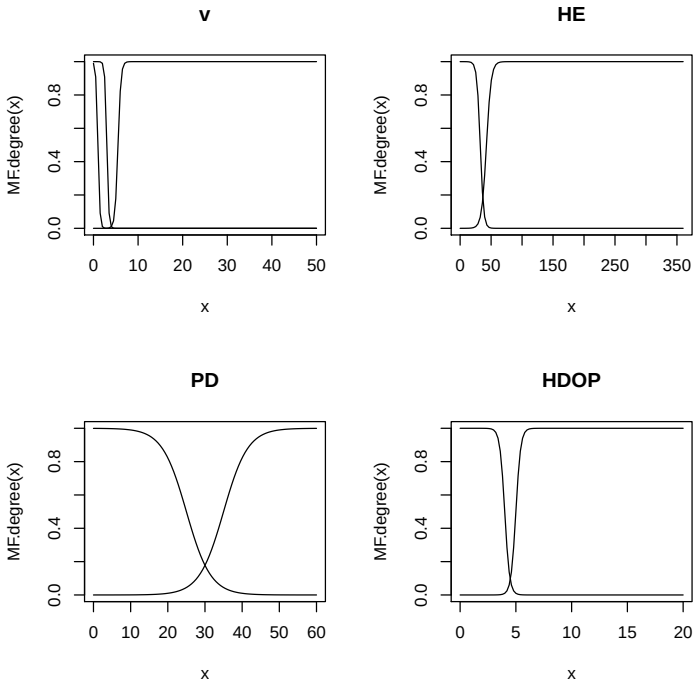

Figure 3: Membership Functions

References

Quddus, Mohammed A. 2006 (January). *High Integrity Map Matching Algorithms for Advanced Transport Telematics Applications*. Ph.D. thesis, Imperial College London, United Kingdom.