Package ‘stdbscan’

January 27, 2026

Title Spatio-Temporal DBSCAN Clustering

Version 0.1.0

Description Implements the ST-DBSCAN (spatio-temporal density-based spatial clustering of applications with noise) clustering algorithm for detecting spatially and temporally dense regions in point data, with a fast C++ backend via 'Rcpp'. Birant and Kut (2007) <doi:10.1016/j.datak.2006.01.013>.

License GPL (>= 3)

Depends R (>= 3.5.0)

Encoding UTF-8

RoxygenNote 7.3.3

URL `https://github.com/MiboraMinima/stdbscan/`, `https://miboraminima.github.io/stdbscan/`

BugReports `https://github.com/MiboraMinima/stdbscan/issues/`

LinkingTo Rcpp

Imports Rcpp

Suggests knitr, rmarkdown, readr, testthat, ggplot2, lubridate, plotly, covr, MetBrewer

VignetteBuilder knitr

LazyData true

NeedsCompilation yes

Author Antoine Le Doeuff [aut, cre] (ORCID: <`https://orcid.org/0009-0008-8807-3816`>)

Maintainer Antoine Le Doeuff <`antoine.ldoeuff@gmail.com`>

Repository CRAN

Date/Publication 2026-01-27 09:20:08 UTC

Contents

geolife_traj *GPS pings from the GeoLife GPS Trajectories dataset*

Description

Extraction of the GeoLife GPS Trajectories dataset. The selected trajectory id is 000-20081023025304.

Data manipulation applied to the raw data :

- Conversion to EPSG:4586
- Manual selection of the pings
- Selection of relevant variables

Usage

```
geolife_traj
```

Format

A `data.frame` with one row per ping and the following columns:

- `date` (chr): The date
- `time` (chr): The time
- `x` (dbl): Longitude (EPSG:4586)
- `y` (dbl): Latitude (EPSG:4586)

Source

`https://www.microsoft.com/en-us/download/details.aspx?id=52367`

Examples

```
data(geolife_traj)
head(geolife_traj)
```

`st_dbscan` *Spatio-Temporal DBSCAN*

Description

Perform **ST-DBSCAN** clustering on points with spatial and temporal coordinates. This algorithm identifies clusters of points that are close both in space and time.

Usage

```
st_dbscan(x, y, t, eps_spatial, eps_temporal, min_pts)
```

Arguments

`x` Numeric vector of x-coordinates (spatial).

`y` Numeric vector of y-coordinates (spatial).

`t` Numeric vector of time values. `t` is expected to represent elapsed time since a common origin (*e.g.* c(0, 6, 10)).

`eps_spatial` Numeric. The spatial radius threshold. Points closer than this in space may belong to the same cluster.

`eps_temporal` Numeric. The temporal threshold. Points closer than this in time may belong to the same cluster.

`min_pts` Integer. Minimum number of points required to form a core point (standard DBSCAN parameter).

Details

ST-DBSCAN extends classical DBSCAN by incorporating a temporal constraint. Two points are considered neighbors if they are within `eps_spatial` in space **and** within `eps_temporal` in time. Clusters are expanded from core points recursively following the DBSCAN algorithm.

This function is implemented in C++ via Rcpp for performance.

Value

An integer vector of length `length(x)` with cluster assignments:

- `-1`: noise point
- `>=1`: cluster ID

References

Birant, D., & Kut, A. (2007). ST-DBSCAN: An algorithm for clustering spatial–temporal data. Data & Knowledge Engineering, 60(1), 208–221. https://doi.org/10.1016/j.datak.2006.01.013

Examples

```
data(geolife_traj)

geolife_traj$date_time <- as.POSIXct(
  paste(geolife_traj$date, geolife_traj$time),
  format = "%Y-%m-%d %H:%M:%S",
  tz = "GMT"
)

geolife_traj$t <- as.numeric(
  geolife_traj$date_time - min(geolife_traj$date_time)
)

st_dbscan(
  x = geolife_traj$x,
  y = geolife_traj$y,
  t = geolife_traj$t,
  eps_spatial = 3, # meters
  eps_temporal = 30, # seconds
  min_pts = 3
)
```

Index