Package ‘seismic’

October 14, 2022

Type Package

Title Predict Information Cascade by Self-Exciting Point Process

Version 1.1

Date 2022-05-20

Author Hera He, Murat Erdogdu, Qingyuan Zhao

Maintainer Qingyuan Zhao <qingyzhao@gmail.com>

Description

An implementation of self-exciting point process model for information cascades, which occurs when many people engage in the same acts after observing the actions of others (e.g. post resharings on Facebook or Twitter). It provides functions to estimate the infectiousness of an information cascade and predict its popularity given the observed history. See <http://snap.stanford.edu/seismic/> for more information and datasets.

URL http://snap.stanford.edu/seismic/

License GPL-3

NeedsCompilation no

RoxygenNote 7.1.2

Repository CRAN

Date/Publication 2022-05-20 21:30:02 UTC

R topics documented:

`get.infectiousness` *Estimate the infectiousness of an information cascade*

Description

Estimate the infectiousness of an information cascade

Usage

```
get.infectiousness(
  share.time,
  degree,
  p.time,
  max.window = 2 * 60 * 60,
  min.window = 300,
  min.count = 5
)
```

Arguments

`share.time`	observed resharing times, sorted, share.time[1] =0
`degree`	observed node degrees
`p.time`	equally spaced vector of time to estimate the infectiousness, p.time[1]=0
`max.window`	maximum span of the locally weight kernel
`min.window`	minimum span of the locally weight kernel
`min.count`	the minimum number of resharings included in the window

Details

Use a triangular kernel with shape changing over time. At time p.time, use a triangluer kernel with slope = min(max(1/(`p.time`/2), 1/`min.window`), `max.window`).

Value

a list of three vectors:

- infectiousness. the estimated infectiousness
- p.up. the upper 95 percent approximate confidence interval
- p.low. the lower 95 percent approximate confidence interval

Examples

```
data(tweet)
pred.time <- seq(0, 6 * 60 * 60, by = 60)
infectiousness <- get.infectiousness(tweet[, 1], tweet[, 2], pred.time)
plot(pred.time, infectiousness$infectiousness)
```

pred.cascade *Predict the popularity of information cascade*

Description

Predict the popularity of information cascade

Usage

```
pred.cascade(
  p.time,
  infectiousness,
  share.time,
  degree,
  n.star = 100,
  features.return = FALSE
)
```

Arguments

`p.time`	equally spaced vector of time to estimate the infectiousness, p.time[1]=0
`infectiousness`	a vector of estimated infectiousness, returned by `get.infectiousness`
`share.time`	observed resharing times, sorted, share.time[1] =0
`degree`	observed node degrees
`n.star`	the average node degree in the social network
`features.return`	if TRUE, returns a matrix of features to be used to further calibrate the prediction

Value

a vector of predicted populatiry at each time in `p.time`.

Examples

```
data(tweet)
pred.time <- seq(0, 6 * 60 * 60, by = 60)
infectiousness <- get.infectiousness(tweet[, 1], tweet[, 2], pred.time)
pred <- pred.cascade(pred.time, infectiousness$infectiousness, tweet[, 1], tweet[, 2], n.star = 100)
plot(pred.time, pred)
```

seismic — *Predicting information cascade by self-exciting point process model*

Description

This package implements a self-exciting point process model for information cascades. An information cascade occurs when many people engage in the same acts after observing the actions of others. Typical examples are post/photo resharings on Facebook and retweets on Twitter. The package provides functions to estimate the infectiousness of an information cascade and predict its popularity given the observed history. For more information, see http://snap.stanford.edu/seismic/.

References

SEISMIC: A Self-Exciting Point Process Model for Predicting Tweet Popularity by Q. Zhao, M. Erdogdu, H. He, A. Rajaraman, J. Leskovec, ACM SIGKDD Conference on Knowledge Discovery and Data Mining (KDD), 2015.

tweet — *An example information cascade*

Description

A dataset containing all the (relative) resharing time and node degree of a tweet. The original Twitter ID is 127001313513967616.

Format

A data frame with 15563 rows and 2 columns

Details

- relative_time_second. resharing time in seconds
- number_of_followers. number of followers

Index