

Quantifying Spatio-Temporal Variation of Invasion Spread: the **invasionSpeed** Package

Joshua Goldstein* Jaewoo Park[†]

Sep 2016

Contents

1 Introduction	1
2 Detecting continuous expansion	2
2.1 Fitting Gaussian process gradient models	2
2.2 Summary and visualization	3
3 Detecting long-range jumps	3
3.1 Rayleigh test	3
3.2 Radial expansion of gradient models	5

1 Introduction

The **invasionSpeed** package provides user-friendly tools in **R** to quantify spatio-temporal variation of spread patterns of invasive species and infectious diseases. The data sets are assumed to contain observations of the time of first appearance and their spatial locations. This package build on Gaussian process gradient models (Goldstein et al., 2018; Banerjee et al., 2003), which are useful for estimating both the speed and direction of continuous expansion, as well as detecting the long-range jumps of such processes. The **invasionSpeed** package implements Bayesian inference for gradient models and provides tools for visualizing inference results. Bayesian inference is carried out via Markov chain Monte Carlo (MCMC) by using functions from the **spBayes** package. This vignette provides a data analysis example using the Hemlock Wolly Adelgid data¹.

*Biocomplexity Institute of Virginia Tech, 900 N Glebe Road Arlington, VA 22203, USA; joshg22@vt.edu

[†]Department of Statistics, Pennsylvania State University, 420 Thomas Building, University Park, PA 16802, USA; jzp191@psu.edu

¹Records are from the US Forest Service Forest Health Protection.

Load the **invasionSpeed** package as follows.

```
> library(invasionSpeed)
```

2 Detecting continuous expansion

`localgrad` function in the **invasionSpeed** package estimates the speed and direction of spread of processes under a Gaussian process framework. The `localgrad` function fits spatial regression models, where a response variable is time to first appearance and the predictors are spatially varying environmental and geographical covariates (e.g. coordinates). If the response surface is the waiting time to first appearance, then the reciprocal of the gradient length is a measure of the invasion speed; fast spread leads to shallow waiting time surfaces, while slow spread results in steep surfaces. For given fitted spatial regression model parameters, the speed and dominant directions of processes are estimated by evaluating gradients of the model.

2.1 Fitting Gaussian process gradient models

Load the `hemlock` data as follows.

```
> data(hemlock)
```

Then, specify longitude and latitude coordinates and years of the first appearance from the `hemlock` data.

```
> coord = cbind(hemlock$long, hemlock$lat)    # longitude latitude coordinates
> dates = hemlock$first.year                 # quarantine data by county
```

To estimate model parameters users can specify the number of batches, batch length, and the acceptance rate, which are tuning parameters for an MCMC sampler. These MCMC tuning parameters come with default values.

```
> amcmc=list(n.batch=10, batch.length=100, accept.rate=0.3)
```

In the `localgrad` function, users can specify time of first appearance (for our woolly adelgid example, this is the year of first appearance), and coordinates associated with each appearance. When the process of interest is on the longitude and latitude domain, users have to specify the 2-dimensional **Albers** equal area projection parameter. If there are additional covariates except for the coordinates of the process, users can also specify `covar`. Then a Bayesian spatial regression model is fitted as follows.

```
> out.grad = localgrad(dates=dates, coord.longlat=coord, Albers=c(29.5, 45.5),
+                       n.samp=1000, amcmc=amcmc)
> class(out.grad)

[1] "localgrad"
```

The `localgrad` function returns an object of class "localgrad".

2.2 Summary and visualization

`summary` function displays the number of locations where there is a statistically significant continuous spread of the process. Mean and median speed of spread are printed on the `km` scale for the process, which is on the longitude and latitude domain.

```
> summary(out.grad)
```

Call:

```
localgrad(dates = dates, coord.longlat = coord, Albers = c(29.5, 45.5), n.samp = 1000, amcn
```

```
Speed of spread estimated at 340 locations.  
Of these, 180 are significantly nonzero.
```

```
Mean speed of spread is estimated as 21.45447 km.
```

```
Median speed of spread is estimated as 13.75958 km.
```

`plotgrad` function draws the vector field plot for the significant gradients.

```
> plotgrad(out.grad, cex=1, pch=".", database="state")
```

In Figure 1, arrows indicate direction and speed of spread. As the speed of spread increases, the length of the arrows becomes longer. The color of the each arrow represents the time of first appearance of the process. Red implies the earliest appearance, and blue indicates the latest one.

3 Detecting long-range jumps

The **invasionSpeed** package identifies candidate locations for long-range jumps in two ways. The first is the Rayleigh test (Jammalamadaka and Sengupta, 2001), and the second is a direct test using Gaussian process gradient models (Goldstein et al., 2018; Banerjee and Gelfand, 2006).

3.1 Rayleigh test

The Rayleigh test is a statistical test of whether a circular distribution is random or non-random. When applied to the gradients of spread near a point, a non-random distribution implies a potential long range spread through that point. `longrange.localgrad` function conducts Rayleigh test to the gradients from the class "localgrad" and points out candidate locations on the map.

```
> plotgrad(out.grad, cex=1, pch=".", database="state")  
> out.longrange = longrange.localgrad(out.grad, add=TRUE)
```

In Figure 2, the black square dots represent candidate locations for long-range jumps. `longrange.localgrad` function tests long range jumps for a given radius

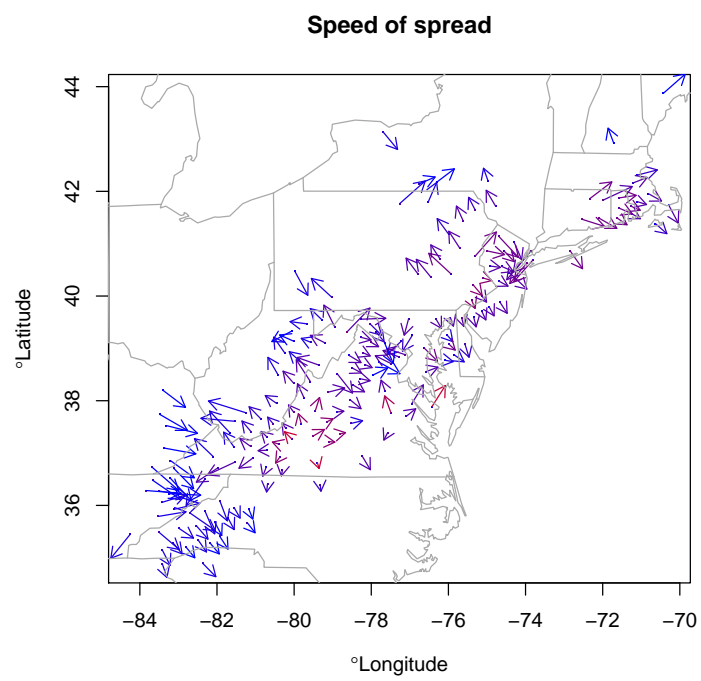


Figure 1: gradient plot for `localgrad` object

Long range jumps tested for 61.84688 km.

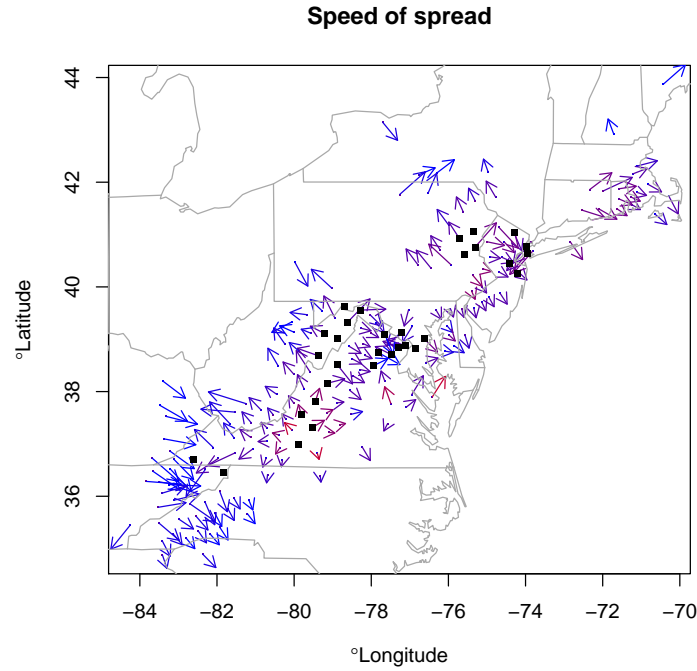


Figure 2: Detecting long-range jump using the Rayleigh test

(61.84688km in this example). User can specify the radius of jumps to perform the Rayleigh test using `r` option in the `longrange.localgrad` function. Otherwise, a default value is used for testing.

3.2 Radial expansion of gradient models

`jump.scan` function directly tests if there is significant radial expansion around a point. For each spatial location, the `jump.scan` function tests the gradient normal on four sides of a square with sides of length `r`. As a heuristic we can flag the location as a potential site of a long-range jump if the spread is significantly out of at least two sides of the box. We can plug in parameter estimates for the Gaussian process gradient model using the posterior samples from class `"localgrad"`.

```
> par.mean = apply(out.grad$post.samp,2,mean)
> out.jump <- jump.scan(dates=dates, coord.longlat=coord,
+                       params = par.mean, Albers= c(29.5,45.5) )
```

Long range jumps tested for 61.84688 km.

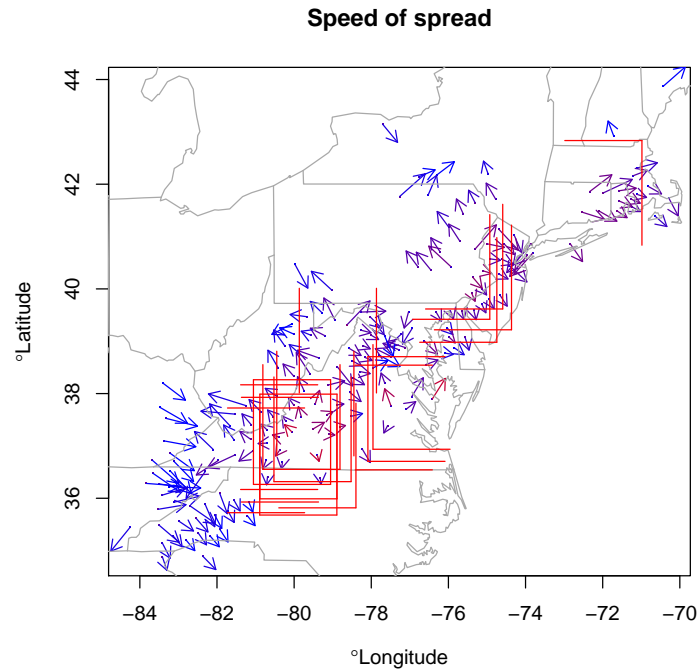


Figure 3: Detecting long-range jump using a Gaussian process gradient model

```
> class(out.jump)
```

```
[1] "jump.scan"
```

The `jump.scan` function returns an object of class "jump.scan". The `plot-boxes` function draws boxes, which represent the detected long-range jumps from class "jump.scan". The `num.sides` option decides the number of significant sides to be drawn.

```
> plotgrad(out.grad,cex=1,pch=".",database="state")
> plotboxes(object = out.jump, num.sides = 2,method="box")
```

We can check candidate locations for long-range jumps (center of the boxes) from Figure 3. In addition, `plotboxes` function prints the tested radius of the jumps (61.84688km in this example). User can specify the radius of jumps using the `r` option in the `jump.scan` function; otherwise a default value is used for testing.

Long range jumps tested for 61.84688 km.

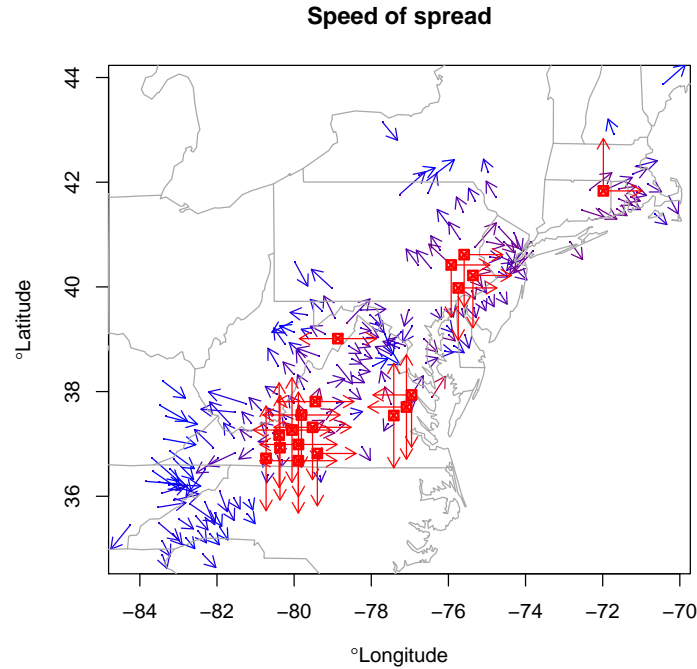


Figure 4: Detecting long-range jump using a Gaussian process gradient model

```
> plotgrad(out.grad,cex=1,pch=".",database="state")  
> plotboxes(object = out.jump, num.sides = 2,method="arrow",point=TRUE)
```

We can draw arrow plots by specifying `method` as `arrow`. In Figure 4, we plot candidate locations and arrows according to the detected direction of long-range jumps.

References

- Banerjee, S., Gelfand, A., and Sirmans, C., 2003, Directional rates of change under spatial process models, *Journal of the American Statistical Association*, 98(464):946-954.
- Banerjee, S. and Gelfand, A., 2006, Bayesian wombling: Curvilinear gradient assessment under spatial process models, *Journal of the American Statistical Association*, 101(476):1487-1501.

- Goldstein, J., Jaewoo Park, Haran, M., Liebhold, A., and Bjornstad, O. N., 2018, Quantifying Spatio-Temporal Variation of Invasion Spread, arXiv preprint arXiv:1506.02685v3.
- Jammalamadaka, S. R. and Sengupta, A., 2001, Topics in circular statistics. World Scientific.